SINFONIA PER UN COMPUTER

VIC 20

Tim Hartnell

EDIZIONE ITALIANA

GRUPPO EDITORIALE JACKSON

SINFONIA PER UN COMPUTER

VIC 20

Tim Hartnell

GRUPPO
EDITORIALE
JACKSON
Via Rosellini, 12
20124 Milano

SOMMARIO

INTRODUZIONE

Il VIC 20 è un computer stimolante, progettato per assicurarti facilmente l'uso migliore del suo colore e potenziale di suono.

È stato molto divertente scrivere questo libro sia per i cooperatori che per i programmatori di video-games del VIC. Il BASIC é veloce e chiaro, la gamma dei suoni e degli effetti cromatici praticamente infinita e, malgrado le lievi limitazioni di una grafica larga, permette di creare splendide composizioni sullo schermo, specialmente usando il comando POKE.

Sebbene io speri che registrerete questi programmi come noi li abbiamo catalogati qui, divertendovi a giocarci, il reale valore di un libro come questo consiste in come maneggerete i programmi per farli vostri nelle idee che ne trarrete e che userete per i vostri prossimi programmi.

Ho tentato di scegliere una vasta gamma di programmi per mostrare quanto possa essere versatile il VIC, e di darvi qualche idea sul tipo di programmi che potrete scrivere e il modo più facile per raggiungere gli effetti desiderati.

La stampante riproduce tutte le grafiche VIC, e per rendere i list più chiari possibili li abbiamo stampati a grandezza naturale, proprio come sono usciti dalla stampante. Malgrado ciò alcuni dei comandi di colore possono essere un po' duri da utilizzare. Questa é una lista dei comandi di colore, insieme alle istruzioni REVERSE ON e REVERSE OFF come riprodotte dalla stampante. Questo dovrebbe aiutarvi nel leggere i list.

Oltre ad usare i comandi come elencati diversi programmi hanno l'istruzione REM che spiega quali colori sono stati usati. Se non capite cos'é un particolare comando di colore usatene uno qualsiasi a piacere, e, in ogni caso cambiate i colori secondo i vostri bisogni, una volta capito cosa si intende fare con un particolare programma.

Ringrazio per il loro utilissimo contributo i programmatori Adam e Lance Burbidge, Crhistopher Huther, M. Kendall, G. Charlton e Ken Mahogany. Senza di loro questo libro non sarebbe riuscito pienamente ad esplorare i potenziali del VIC 20.

Il libro segue un piano lineare. Viene introdotto il programma, qualche informazione su come fu scritto e come raggiunge i suoi scopi. Viene anche dato qualche spunto su come potresti modificarlo. Oltre ai list abbiamo aggiunto qualche idea per aiutarti a scrivere i tuoi programmi e ci sono alcuni listati di altri programmi in Microsoft BASIC che tu puoi voler adattare al VIC. La maggior parte dei programmi che tu incontri nei libri e nelle riviste

```
1 REM CARATTERI DI CONTROLLO
10 REM "■" NERO
15 REM
20 REM "■" BIANCO
25 REM
30 REM "■" ROSSO
35 REM
40 REM "◣" AZZURRO
45 REM
50 REM "■" PORPORA
55 REM
60 REM "■" VERDE
65 REM
70 REM "■" BLU
75 REM
80 REM "■" GIALLO
85 REM
90 REM "■" REVERSE ON
95 REM
100 REM"■" REVERSE OFF
105 REM
110 REM"■" IN ALTO
115 REM
120 REM"■" IN BASSO
125 REM
130 REM"■" A DESTRA
135 REM
140 REM"■" A SINISTRA
READY.
```

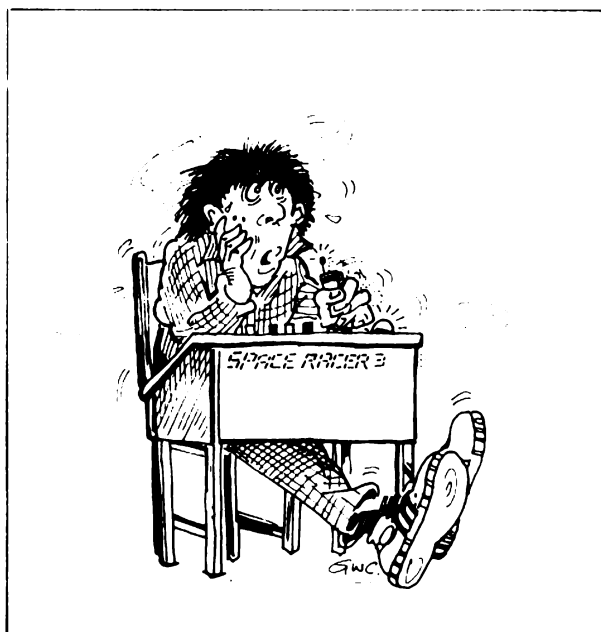
sono scritti, come ben sai, in linguaggio BASIC diverso dal VIC BASIC, così il convertire questi programmi potrà aiutarti a sviluppare l'abilità necessaria a convertire i list che troverai da altre parti.

Infine c'é un'appendice, intesa a servire da prontuario, con un insieme di informazioni a cui sei sicuro di poterti sempre riferire. Sto già lavorando ad

un secondo volume; saranno quindi benaccetti i vostri contributi, sia programmi che scorciatoie e routines per effetti speciali scoperti da voi. Se accetterò i vostri contributi riceverete un compenso, il riconoscimento della paternità e alcune copie del libro. Spedite i list e le descrizioni dettagliate a:
INTERFACE 44–66 EARLS COURT ROAD, LONDON, W8 6EJ

Bene, sono sicuro che sarete ansiosi di passare ai fatti, termino così questa introduzione e vi lascio cominciare. Prima, però, vorrei ringraziare la ADDA HOME COMPUTERS Ltd, Victoria Road, London W3 che mi ha molto aiutato fornendo l'apparecchiatura e utili informazioni.

Spero che vi divertirete usando questo libro. Mi interessa qualunque risposta vi preoccuperete di mandarmi (scrivetemi all'INTERFACE) e attendo impazientemente di ricevere i vostri programmi originali preferiti, le routine e tante idee per il nostro secondo volume. Allo stesso modo gradirò molto ricevere una qualsiasi ristesura o miglioramento dei programmi di questo libro (per favore solo listati). Buon gioco!

TIM HARTNELL

PAROLIERE

È questo un gioco per due persone, la prima inserisce una parola di sua scelta (lettera per lettera) e l'altra cerca di indovinare tale parola (una lettera alla volta). Il VIC stampa una barretta per ogni lettera della parola ed ogni volta che una lettera sarà indovinata questa verrà stampata al posto di ogni barretta che, all'interno della parola, la rappresenta.

Remarks

50	Crea degli array, se necessario
100-130	accetta la parola "segreta", una lettera alla volta
135	conta i 10 tentativi
145-170	i colori vengono generati casualmente secondo la subroutine 9010, stampa la lettera se è stata indovinata, altrimenti stampa la barretta
210-260	accetta una lettera e la confronta, lettera per lettera, con la parola "segreta"
270	se la parola viene indovinata il controllo è passato alla routine di vittoria (linea 1000)
300-330	messaggio di sconfitta e rivelazione della parola
1055-1100	chiede se si vuole giocare ancora, se la risposta è no allora ringrazia per la partita
5000	routine che provvede al suono
9000	routine che assegna i controlli colore alla stringa A$, quando viene chiamata (gosub 9010) stampa il prossimo carattere con un colore random. Questa routine chiama anche quella del suono, opera poi dei flashes dando al bordo dei colori casuali

```
 10 REM PAROLIERE (RIVISTO DA COMINI R.
)
 20 PRINT"■":GOSUB9000
 30 INPUT"■■■QUANTE LETTERE";N
 50 IFN>10THENDIMB(N),D(N)
 100 FORA=1TON:INPUTC$
 120 B(A)=ASC(C$):D(A)=B(A)
 125 C$="."
 130 NEXTA
 135 FORJ=1TO10:PRINT"■"
 140 PRINT:PRINT:PRINT
 145 FORE=1TON
 150 GOSUB9010:PRINTB$;
 155 IFB(E)=D(E)THENPRINT"-";
 160 IFB(E)<>D(E)THENPRINTCHR$(B(E));
 170 NEXTE
 180 PRINT:PRINT
 190 H=0
 200 GOSUB9010:PRINTB$; "■■■■■TENTATIVO
"J
 210 INPUTC$
 220 F=ASC(C$)
 230 FORG=1TON
 240 IFD(G)=FTHEND(G)=0
 250 IFD(G)=0THENH=H+1
 260 NEXTG
 270 IFH=NTHEN1000
 280 NEXTJ
 290 PRINT
 300 GOSUB9010:PRINT"SPIACENTE IL TEMPO
 E'■■■■■■■■SCADUTO"
 310 PRINT
 320 GOSUB9010:PRINTB$;" LA PAROLA ERA
";
 330 GOTO1045
 1000 REM VITTORIA
 1010 PRINT
 1020 GOSUB9010:PRINTB$;"  HAI INDOVINA
TO IN■■■■■■■■■■■■"J"■■■■■■■■TENTATIVI■"
 1030 PRINTCHR$(28)"■■■■LA PAROLA ERA■"
 1040 PRINT"■■■■■■■■";
 1045 FORQ=1TON:PRINTCHR$(B(Q));:NEXT
 1050 PRINT
```

```
 1055 GOSUB9010:PRINTB$;"█████GIOCHI A
NCORA███████████(S/N)"
 1060 GETC$:IFC$=""THEN1060
 1070 IFC$="S"THENRUN
 1080 GOSUB9010:PRINTB$;"█████OK, GRAZ
IE PER"
 1090 GOSUB9010:PRINTB$,"█████LA PARTIT
A█"
 1095 FORT=1TO300:NEXT
 1097 PRINT
 1100 GOTO1080
 5000 REM SUONO
 5010 POKE36878,INT(RND(1)*8)+8
 5020 FORT=1TO100STEP(INT(RND(1)*8)+1)
 5030 POKE36876,240-T
 5040 POKE36877,T
 5060 NEXT
 5070 POKE36876,0
 5080 POKE36877,0
 5090 RETURN
 8990 END
 9000 A$="███████"
 9010 B$=MID$(A$,(INT(RND(1)*7)+1),1)
 9020 GOSUB5000
 9025 POKE36879,INT(RND(1)*6)+26
 9030 RETURN
```

Lista simboli grafici

```
  20 : 1  SHIFT HOME =CHR$(147)

  30 : 3  CRSR← =CHR$(29)

 135 : 1  SHIFT HOME =CHR$(147)

 200 : 5  CRSR← =CHR$(29)

 300 : 8  CRSR← =CHR$(29)
```

```
1020 : 1   CRSR↑ =CHR$(17)
       10   SHIFT CRSR← =CHR$(157)
       1   CRSR↑ =CHR$(17)
       7   SHIFT CRSR← =CHR$(157)
       1   CRSR↑ =CHR$(17)

1030 : 4   CRSR← =CHR$(29)
       1   CRSR↑ =CHR$(17)

1040 : 8   CRSR← =CHR$(29)

1055 : 2   CRSR↑ =CHR$(17)
       1   REVERSE =CHR$(18)
       3   CRSR← =CHR$(29)
       1   SHIFT REVERSE =CHR$(146)
       1   CRSR↑ =CHR$(17)
       9   SHIFT CRSR← =CHR$(157)

1080 : 1   REVERSE =CHR$(18)
       5   CRSR← =CHR$(29)

1090 : 1   REVERSE =CHR$(18)
       4   SHIFT CRSR← =CHR$(157)
       1   SHIFT REVERSE =CHR$(146)
```

CORSE

Questo gioco è molto semplice. Scegliete il vostro "cavallo", un numero tra 1 e 6, poi non vi resta che osservare la corsa e vedere se le vostre previsioni erano esatte. Interessanti sono le linee 15 e 55. Queste permettono al VIC di assegnare un colore diverso ad ogni numero pur mantenendo per ogni numero sempre lo stesso colore ogni volta che viene stampato. Capirete meglio cosa intendo quando vedrete girare il programma.

La routine del suono, linea 1000 potrà facilmente essere migliorata dalla vostra abilità di programmatori.

```
 5 POKE36879,8
 10 REM CORSE (RIVISTO DA COMINI R.)
 12 PRINT"■":PRINT:PRINT:PRINT:PRINT:PR
INT:PRINT
 15 A$="■■■■■■"+CHR$(28)
 20 PRINT"■■■NOME VINCITORE",,,,,,,CHR$
(28)"■■■■■(DA 1 A 6)";
 30 INPUTW
 35 IFW<1ORW>6THEN30
 40 FORJ=1TO6:B(J)=0:NEXT
 50 PRINT"■"
 52 PRINT"■":REM HOME
 54 FORM=1TO6
 55 C$=MID$(A$,M,1)
 60 PRINTTAB(B(M));C$   ;M
 65 GOSUB1000
 70 PRINT
 80 B(M)=B(M)+INT(RND(1)+.5)
 90 IFB(M)>18THEN120
 100 NEXT
 110 GOTO52
 120 FORZ=MTO8
```

```
 130 PRINT
 140 NEXT
 150 PRINT"    FINE GIOCO      "
 160 PRINT
 170 PRINTCHR$(28)"IL VINCITORE E´"M
 180 IFW=MTHENPRINT"E TU LO AVEVI DET
TO!!":PRINT"MOLTO BENE"
 185 PRINT
 190 IFW<>MTHENPRINT"AVEVI DETTO"W
:PRINT:PRINT"PECCATO"
 200 PRINT
 210 FORJ=1TO800:NEXT
 220 PRINT"PREMI UN TASTO"
 230 PRINTCHR$(28)"PER"
 240 PRINT"UNA NUOVA PARTITA"
 250 GETV$
 260 IFV$=""THEN250
 270 RUN
 1000 POKE36878,15
 1010 FORQ=1TOA(M)
 1020 POKE36876,180+INT(RND(1)*60)
 1030 NEXTQ
 1040 POKE36878,0
 1070 RETURN
```

Lista simboli grafici

```
 12 : 1  SHIFT HOME =CHR$(147)

 20 : 8  CRSR← =CHR$(29)

 50 : 1  SHIFT HOME =CHR$(147)

 52 : 1  HOME =CHR$(19)

150 : 1  REVERSE =CHR$(18)
      1  CRSR← =CHR$(29)

170 : 1  REVERSE =CHR$(18)
      2  CRSR← =CHR$(29)
```

```
180 : 6  CRSR← =CHR$(29)

190 : 10  CRSR← =CHR$(29)

220 : 3  CRSR← =CHR$(29)

230 : 8  CRSR← =CHR$(29)

240 : 2  CRSR← =CHR$(29)
```

DR WATSON e DR AUDIO

Questa è una coppia di semplici programmi nei quali il Vic pensa un numero tra uno e cento e voi dovete indovinarlo.

In DR WATSON l'aiuto del VIC è verbale ed è graziosamente facile indovinare il numero. DR AUDIO, di contro, dà un aiuto "musicale", cioè una nota, che vi permetterà di capire a che distanza siete dal numero da indovinare. Dopo avervi giocato due o tre volte probabilmente sarete più che soddisfatti di vedere come vi risulti più semplice l'interpretazione dei suoni.

Dr Watson

```
 5 REM DOTTOR WATSON
 6 C=0:PRINT"■"
 10 PRINT"STO PENSANDO UN NUMERO     TRA
 UNO E 100"
 30 A=INT(RND(1)*100)+1
 40 PRINT"QUAL'E";
 45 INPUTB
 50 C=C+1
 60 IFB<>ATHENPRINT"  NO,"CHR$(28);B;"■
E'"CHR$(28)" SBAGLIATO"
 65 IFB=ATHEN120
 70 IFB<ATHENPRINT"■   E' TROPPO BASSO■
"
 80 IFB>ATHENPRINT"■   E' TROPPO ALTO■"
 90 IFC<10THEN45
 100 PRINT:PRINT"    ■FINE DEL GIOCO■"
 110 PRINTCHR$(28)"IO STAVO PENSANDO";A
:GOTO170
 120 PRINT"     SI,E' GIUSTO!"
 130 PRINT
 140 PRINT"■  IO AVEVO PENSATO"A
 150 PRINT
```

```
 160 PRINT" HAI INDOVINATO IN "C"
TENTATIVI"
 170 FORJ=1TO5000:NEXT
 180 RUN
```

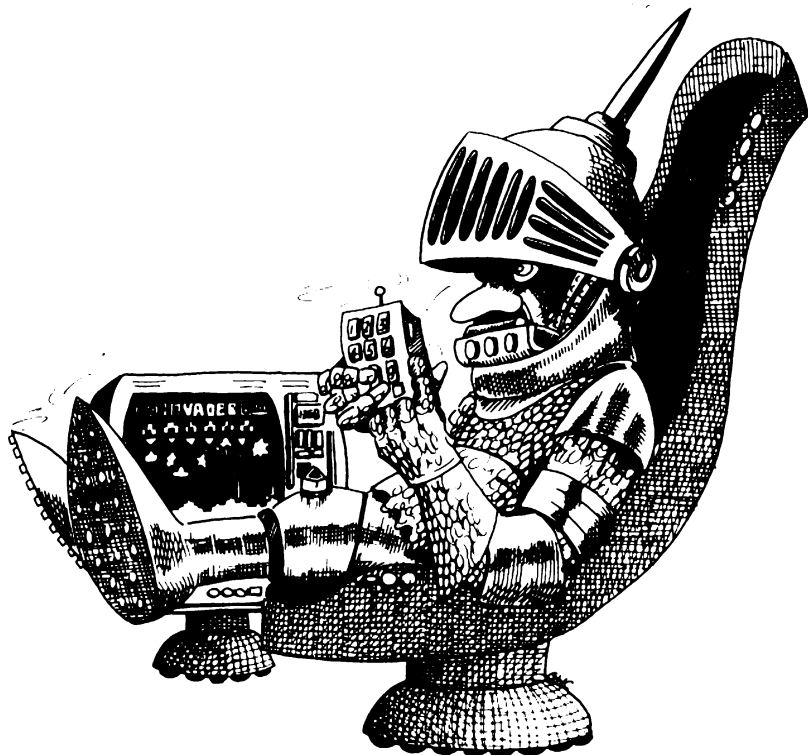
```
Lista simboli grafici

   6  :  1   SHIFT HOME =CHR$(147)

 100  :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
```

Dr Audio

```
 5 REM DR AUDIO
 6 C=1
 7 POKE36878,7
 10 PRINT"■STO PENSANDO UN NUMERO
 TRA 1 E 100"
 30 A=INT(RND(1)*100)+1
 40 PRINT"■■■■■■QUAL'E'";
 45 INPUTB
 50 C=C+1
 60 IFB<>ATHENPRINT"NO,■";B;"■E' ■SBAGL
IATO"
 65 IFB=ATHEN120
 67 PRINT"TENTATIVO";C;
 70 POKE36875,ABS(A-B)+128
 80 FORT=1TO700:NEXTT
 85 POKE36875,0
 90 IFC<30THEN45
 100 PRINT:PRINT"    ■■FINE DEL GIOCO■■"
 110 PRINTCHR$(28)"IO STAVO PENSANDO"A:
GOTO170
 120 PRINT"    ■SI E' GIUSTO!■"
 130 PRINT
 140 PRINT"■ IO STAVO PENSANDO■"A"■"
 150 PRINT
```

```
160 PRINT" HAI INDOVINATO IN";C"
TENTATIVI"
170 FORJ=1TO5000:NEXT
180 RUN
```

Lista simboli grafici

```
 10 : 1   SHIFT HOME =CHR$(147)

 40 : 1   CRSR↑ =CHR$(17)
      5   CRSR← =CHR$(29)

100 : 1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
```

MASTERMIND

Il gioco Mastermind è registrato dalla Invicta, che ne vende molte versioni facenti uso di chiodi colorati. L'Invicta comprò i diritti del gioco da un matematico, Mordechai Meirovich, nel 1971 e dopo un paio d'anni lo commercializzò con grande successo. Il principio è semplice. Il VIC sceglie un codice di tre cifre (come ad esempio 259), usando numeri da 1 a 9. Lo zero non è usato e nessuna cifra è usata più di una volta nel codice (così 117, 494, o 666 sono impossibili). Bisogna indovinare il codice in 10 tentativi (al massimo). La risposta del VIC è facile da interpretare. Dopo ogni tentativo viene segnalato il punteggio in termini di bianchi e neri. Realizzare un nero significa azzeccare una cifra giusta ed al posto giusto all'interno del codice. Realizzare un bianco significa aver azzeccato una cifra del codice ma al posto sbagliato.
Es.: se il codice è 217 inserendo 261 si otterrà un "punteggio' di un nero ed un bianco, il nero si riferisce al 2 ed il bianco all'1. Se dopo 10 tentativi non avrete indovinato il codice questo vi sarà mostrato dal VIC.

Remarks

30-50	vengono generati i tre numeri del codice
60	si controlla che non vi siano numeri uguali e se ve ne sono si torna a generarne tre nuovi
70	le tre cifre separate vengono unite in un numero di tre cifre
80 e 330	ciclo dei tentativi
90	stampa il numero del tentativo
95	viene accettato il codice dato dal giocatore
100-120	il codice fornito dal giocatore come un unico numero viene separato nelle tre cifre componenti, che possono così essere comparate con quelle generate dal VIC

130	comparazione del codice del giocatore con quello del VIC, se sono uguali si va alla routine di vincita (linea 360)
150	vengono azzerati i contatori dei neri e dei bianchi
170-210	controlla i neri. Per ogni cifra giusta al posto giusto il contatore dei neri è incrementato di uno (linea 190), e viene cambiata con zero la cifra del codice originale affinchè non sia poi ritrovata dalla routine che controlla i bianchi
220-280	controlla i bianchi, per ogni cifra giusta il contatore dei bianchi viene incrementato di uno
290	viene ricostruito il codice originale (tramite D) mettendo così a posto quelle cifre che erano state trasformate in zeri alla linea 200
320-325	stampa del punteggio
340-350	se il giocatore non indovina in 10 tentativi il codice gli viene rivelato
360-380	routine di vincita

Suggerimenti

- ridurre il numero dei tentativi (6 è un buon numero)
- cambiare il gioco con i colori, usando per questi un codice numerico, utile sia per la stampa che per l'input del codice da parte del giocatore
- aggiungere un suono ed un particolare affetto a seconda dei bianchi e dei neri trovati
- aggiungere un effetto vistoso se il giocatore indovina il codice

```
1 REM MASTERMIND
2 REM RIVISTO DA COMINI R.
10 PRINT"[CLR]"
30 FORZ=1TO3
40 A(Z)=INT(RND(1)*9)+1
50 NEXTZ
60 IFA(1)=A(2)ORA(1)=A(3)ORA(2)=A(3)TH
EN30
```

```
70 D=100*A(1)+10*A(2)+A(3)
80 FORC=1TO10
90 PRINT"    TENTATIVO ";C;
95 INPUTX
100 B(1)=INT(X/100)
110 B(2)=INT((X-100*B(1))/10)
120 B(3)=X-100*B(1)-10*B(2)
130 IFD=XTHEN360
150 N=0:W=0
170 FORE=1TO3
180 IFA(E)<>B(E)THEN210
190 N=N+1
200 A(E)=0
210 NEXTE
220 FORF=1TO3
230 IFA(F)=0THEN280
240 FORE=1TO3
250 IFB(F)<>A(E)THEN270
260 W=W+1
270 NEXTE
280 NEXTF
290 A(1)=INT(D/100)
300 A(2)=INT((D-100*A(1))/10)
310 A(3)=D-100*A(1)-10*A(2)
320 PRINT"■   HAI REALIZZATO"
325 PRINT"■";N;"NERI E■";W;"BIANCHI■"
330 NEXTC
340 PRINT"■■■IL MIO NUMERO ERA■■■■■■■■■
■■■■■■■";A(1);A(2);A(3)
350 END
360 PRINTCHR$(28)"■■■CONGRATULAZIONI"
370 PRINT"■■■■■TENTATIVI■■";C"■■"
380 END
```

Lista simboli grafici

```
10  : 1  SHIFT HOME =CHR$(147)

340 : 2  CRSR← =CHR$(29)
      1  CRSR↑ =CHR$(17)
      13  SHIFT CRSR← =CHR$(157)
```

```
360 : 3  CRSR← =CHR$(29)

370 : 4  CRSR← =CHR$(29)
      2  REVERSE =CHR$(18)
```

NOTTE SULLA MONTAGNA SELVAGGIA

Questo programma va forse bene per quelle persone che dicono "é veramente piacevole avere un Personal Computer, ma per cosa lo puoi usare?". Una volta visto (e udito) questo programma, potranno essere convinti che forse non avrebbero dovuto porre tale domanda. La musica è selvaggia e casuale come dice il titolo, sebbene la grafica abbia un ordine e una simmetria. Inserito il programma, come qui è listato, provatelo e se volete cercate poi di dare un ordine a questa musica.

```
 5 PRINT"♥"
 10 REM NOTTE SULLA MONTAGNA SELVAGGIA
 12 REM MUSICA CASUALE
 15 M=INT(RND(1)*8)
 20 A=INT(RND(1)*18)
 30 B=INT(RND(1)*18)
 40 C=7911
 50 D=38631
 60 POKEC+A,102:POKED+A,M
 70 POKEC-A,102:POKED-A,M
 80 POKEC+11*A+B,102:POKED+11*A+B,M
 90 POKEC-11*A-B,102:POKED-11*A-B,M
 97 IFRND(1)>0.3THEN120
 100 POKE36879,16*(INT(RND(1)*16))+INT(
RND(1)*8)+8
 120 POKE36878,INT(RND(1)*15)+1
 130 POKE36874,INT(RND(1)*128)+128
 140 POKE36875,INT(RND(1)*128)+128
 150 POKE36876,INT(RND(1)*128)+128
 180 POKE36877,INT(RND(1)*128)+128
 195 IFRND(1)>.1THEN205
 200 POKE36878,0
 205 IFRND(1)>.3THEN215
 215 IFRND(1)>.3THEN230
```

```
220 POKE36875,0
230 FORT=1TORND(1)*20+5:NEXT
235 IFRND(1)>.3THEN250
240 POKE36877,0
250 POKE36876,0
500 GOTO10
```

Lista simboli grafici

```
5 : 1   SHIFT HOME =CHR$(147)
```

SLOT

Questo splendido programma, che utilizza al massimo la grafica del VIC, è stato scritto per questo libro da M. Kendall.

Ci sono tre ruote che girano, ogni ruota ha quattro simboli. Voi potete fermare tutte le ruote e dopo tenerle come sono o dare ancora dei colpetti con la speranza di "accomodarle" meglio.

Ogni giocata ha naturalmente il suo costo, tale costo è di 2$, voi partite con 5$ a disposizione.

Le combinazioni vincenti

- stesso stesso stesso 5$
- stesso stesso altro 3$
- altro stesso stesso 1$

Tasti di controllo

Ogni tasto ferma le ruote
Con H le tenete come sono
Con N date un colpetto

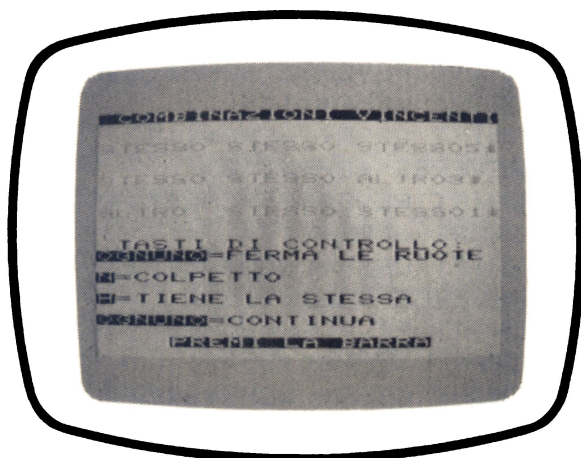
ogni tasto vi fa ripartire

```
 0 POKE36879,27:PRINT"SLOT":P
RINT" DI M.KENDALL 1982":REM RIVISTO DA
 COMINI R.
 1 PRINT"              VUOI LE ":PRINT"
    ISTRUZIONI?":PRINT"              (S
OR N)
 2 GETQ$:IFQ$=""THEN2
 3 IFQ$="S"THENGOSUB1000
 4 L=5
 5 PRINT""
 6 L=L-2
 7 W=0
```

```
8 POKE36878,10
9 POKE36879,72
10 A$(1)="                                        "
20 A$(2)="                                        "
30 A$(3)="                                        "
40 A$(4)="                                        "
45 FORN=1TO3
50 A=INT(RND(1)*4)+1
51 A(N)=A
54 G=(N-1)*7+1
55 PRINTTAB(G)"                                        "
56 PRINT" "
60 PRINTTAB(G);A$(A)
61 POKE36875,200:FORZ=1TO10:NEXT:POKE36875,0
66 PRINT" "
70 NEXTN
80 GETQ$:IFQ$<>""THEN190
100 GOTO45
190 FORN=1TO3
200 GETQ$:IFQ$=""THEN200
210 IFQ$="H"THEN250
220 IFQ$="N"THENGOSUB300:GOTO240
230 GOTO200
240 PRINTTAB((N-1)*7+1)"NUDGE"
245 GOTO260
250 PRINTTAB((N-1)*7+1)"HOLD"
260 NEXTN
270 GOTO500
299 END
300 A=INT(RND(1)*4)+1
301 A(N)=A
302 C$="                                        "
```

```
 304 G=((N-1)*7+1)
 305 PRINTTAB(G)C$
 310 PRINT"";TAB((N-1)*7+1)A$(A)
 320 PRINT""
 330 RETURN
 500 PRINT"    VINCITA";
 510 IFA(1)=A(2)ANDA(2)=A(3)THENW=5:GOT
O550
 520 IFA(1)=A(2)THENW=3
 530 IFA(2)=A(3)THENW=1
 550 PRINTW;"$ MENO 2$ "
 560 REM PRINT"PER QUESTO GIOCO"
 563 IFW=5THENPRINT"         JACKPOT
":GOSUB9000
 565 PRINT""
 570 L=L+W:PRINT" TI RESTANO --";L"$"
 575 IFL>20THEN8000
 580 IFL<0THENPRINT"TU SEI ROVINA
TO!!":GOTO700
 600 GETQ$:IFQ$<>""THEN5
 610 GOTO600
 700 PRINT"     QUESTA PER TE 
E' LA FINE"
 701 FORZ=128TO255:POKE36877,Z:FORT=1TO
10:NEXTT,Z
 702 POKE36877,0
 710 GETQ$:IFQ$=""THEN710
 730 RUN
 1000 PRINT""
 1010 PRINT"   VIC SLOT MACHINE
 1020 PRINTCHR$(28)"
 1030 PRINT"     CI SONO 3 RUOTE
 1040 PRINT" 4 SIMBOLI PER RUOTA
 1060 PRINT"  PUOI FERMARLE TUTTE
 1065 PRINT"
 1080 PRINT"  TENERLE COME SONO

 1085 PRINT"DARE UN COLPETTO.
 1090 PRINTCHR$(28)"  TU PARTI CON   5$

 1100 PRINT"  UN GIOCO COSTA 2$
 1110 PRINT"PREMI LA BARRA"
 1120 GETA$:IFA$=""THEN1120
```

```
 2000 PRINT""
 2019 PRINTCHR$(28)" COMBINAZIONI VINC
ENTI"
 2030 PRINT"STESSO STESSO STESSO5$
 2040 PRINT"STESSO STESSO ALTRO3$
 2050 PRINT"ALTRO  STESSO STESSO1$
 2060 PRINTCHR$(28)" TASTI DI CONTROLLO
:
 2070 PRINT"OGNUNO=FERMA LE RUOTE
 2080 PRINT"N=COLPETTO
 2090 PRINT"H=TIENE LA STESSA
 2100 PRINT"OGNUNO=CONTINUA
 2110 PRINT"      PREMI LA BARRA"
 2120 GETA$:IFA$=""THEN2120
 2130 RETURN
 7999 END
 8000 PRINT"     CONGRATULAZIONI":PRI
NT"HAI VUOTATO LA SLOT":PRINT"NON H
O PIU' SOLDI
 8001 POKE36878,15
 8002 POKE36879,24
 8005 PRINT"QUESTA PER TE E'LA FINE"
 8010 FORN=1TO12
 8015 READA:POKE36875,A
 8020 FORZ=1TO150:NEXTZ
 8030 NEXTN
 8040 DATA135,147,135,175,175,163,183,1
83,191,195,195,195,27
 8100 RESTORE
 8115 READA:POKE36876,A
 8120 FORZ=1TO150:NEXTZ
 8130 NEXTN
 8135 PRINT""
 8140 POKE36875,0
 8150 END
 9000 POKE36878,10
 9001 FORZ=1TO10
 9002 POKE36879,74
 9010 FORM=200TO220+Z*2
 9020 POKE36876,M
 9030 NEXTM,Z
 9035 POKE36876,0
 9036 POKE36879,72
```

```
9040 RETURN
9500 REM-----------------
9510 REM
9520 REM M. KENDALL
9530 REM
9540 REM (C)    1982
9550 REM
9560 REM-----------------
```

Lista simboli grafici

```
0  : 1   SHIFT HOME =CHR$(147)
     1   CRSR↑ =CHR$(17)
     7   CRSR← =CHR$(29)

1  : 9   CRSR↑ =CHR$(17)

5  : 1   SHIFT HOME =CHR$(147)

10 : 1   CRSR↑ =CHR$(17)
     3   SHIFT CRSR← =CHR$(157)
     1   CRSR↑ =CHR$(17)
     4   SHIFT CRSR← =CHR$(157)
     1   REVERSE =CHR$(18)
     1   CRSR↑ =CHR$(17)
     5   SHIFT CRSR← =CHR$(157)
     1   SHIFT REVERSE =CHR$(146)
     1   REVERSE =CHR$(18)
     1   SHIFT REVERSE =CHR$(146)
     1   CRSR↑ =CHR$(17)
     6   SHIFT CRSR← =CHR$(157)

20 : 1   CRSR↑ =CHR$(17)
     6   SHIFT CRSR← =CHR$(157)
     1   REVERSE =CHR$(18)
     1   SHIFT REVERSE =CHR$(146)
     1   CRSR↑ =CHR$(17)
     6   SHIFT CRSR← =CHR$(157)
     1   REVERSE =CHR$(18)
     1   SHIFT REVERSE =CHR$(146)
     1   CRSR↑ =CHR$(17)
     7   SHIFT CRSR← =CHR$(157)
     1   REVERSE =CHR$(18)
     1   SHIFT REVERSE =CHR$(146)
     1   CRSR↑ =CHR$(17)
     7   SHIFT CRSR← =CHR$(157)
     1   REVERSE =CHR$(18)
     1   SHIFT REVERSE =CHR$(146)
```

```
30 : 1   CRSR↑ =CHR$(17)
     4   SHIFT CRSR← =CHR$(157)
     1   CRSR↑ =CHR$(17)
     4   SHIFT CRSR← =CHR$(157)
     1   REVERSE =CHR$(18)
     2   SHIFT REVERSE =CHR$(146)
     1   CRSR↑ =CHR$(17)
     8   SHIFT CRSR← =CHR$(157)
     1   REVERSE =CHR$(18)
     1   SHIFT REVERSE =CHR$(146)


40 : 1   CRSR↑ =CHR$(17)
     5   SHIFT CRSR← =CHR$(157)
     1   REVERSE =CHR$(18)
     1   SHIFT REVERSE =CHR$(146)
     1   REVERSE =CHR$(18)
     1   CRSR↑ =CHR$(17)
     4   SHIFT CRSR← =CHR$(157)
     1   SHIFT REVERSE =CHR$(146)
     1   CRSR↑ =CHR$(17)
     6   SHIFT CRSR← =CHR$(157)
     1   REVERSE =CHR$(18)
     1   SHIFT REVERSE =CHR$(146)
     1   REVERSE =CHR$(18)
     1   CRSR↑ =CHR$(17)
     8   SHIFT CRSR← =CHR$(157)
     1   SHIFT REVERSE =CHR$(146)
     2   CRSR← =CHR$(29)


55 : 7   SHIFT CRSR← =CHR$(157)
     2   CRSR↑ =CHR$(17)
     7   SHIFT CRSR← =CHR$(157)
     1   CRSR↑ =CHR$(17)
     7   SHIFT CRSR← =CHR$(157)
     1   CRSR↑ =CHR$(17)
     7   SHIFT CRSR← =CHR$(157)


56 : 1   HOME =CHR$(19)
```

```
 66  :  1   HOME  =CHR$(19)

240  :  7   CRSR↑  =CHR$(17)
        7   SHIFT CRSR↑  =CHR$(145)

250  :  7   CRSR↑  =CHR$(17)
        7   SHIFT CRSR↑  =CHR$(145)

302  :  1   SHIFT CRSR↑  =CHR$(145)
        1   CRSR↑  =CHR$(17)
        7   SHIFT CRSR←  =CHR$(157)
        1   CRSR↑  =CHR$(17)
        7   SHIFT CRSR←  =CHR$(157)
        1   CRSR↑  =CHR$(17)
        7   SHIFT CRSR←  =CHR$(157)
        1   CRSR↑  =CHR$(17)
        7   SHIFT CRSR←  =CHR$(157)

310  :  1   HOME  =CHR$(19)
        1   CRSR↑  =CHR$(17)

320  :  1   HOME  =CHR$(19)

500  :  1   HOME  =CHR$(19)
        11   CRSR↑  =CHR$(17)

550  :  1   SHIFT CRSR←  =CHR$(157)

560  :  1   CRSR↑  =CHR$(17)
        3   CRSR←  =CHR$(29)

563  :  1   CRSR↑  =CHR$(17)
        1   REVERSE  =CHR$(18)
        1   SHIFT REVERSE  =CHR$(146)
        2   SHIFT CRSR↑  =CHR$(145)

565  :  3   CRSR↑  =CHR$(17)

570  :  1   CRSR↑  =CHR$(17)

580  :  3   SHIFT CRSR↑  =CHR$(145)
        3   CRSR←  =CHR$(29)
```

```
 700 : 2   CRSR↑ =CHR$(17)
       12   SHIFT CRSR← =CHR$(157)

1000 : 1   SHIFT HOME =CHR$(147)

1030 : 1   CRSR↑ =CHR$(17)

1040 : 1   CRSR↑ =CHR$(17)

1065 : 10   CRSR← =CHR$(29)

1080 : 1   CRSR↑ =CHR$(17)
       9   SHIFT CRSR← =CHR$(157)

1085 : 2   CRSR← =CHR$(29)
       1   CRSR↑ =CHR$(17)

1090 : 1   CRSR↑ =CHR$(17)

1110 : 2   CRSR↑ =CHR$(17)
       3   CRSR← =CHR$(29)

2000 : 1   SHIFT HOME =CHR$(147)

2019 : 1   REVERSE =CHR$(18)
       1   SHIFT REVERSE =CHR$(146)
       1   CRSR↑ =CHR$(17)

2030 : 1   CRSR↑ =CHR$(17)

2040 : 2   CRSR↑ =CHR$(17)

2050 : 1   CRSR↑ =CHR$(17)

2070 : 1   REVERSE =CHR$(18)
       1   SHIFT REVERSE =CHR$(146)
       1   CRSR↑ =CHR$(17)

2080 : 1   REVERSE =CHR$(18)
       1   SHIFT REVERSE =CHR$(146)
       1   CRSR↑ =CHR$(17)
```

```
2090 : 1  REVERSE =CHR$(18)
       1  SHIFT REVERSE =CHR$(146)
       1  CRSR↑ =CHR$(17)

2100 : 1  REVERSE =CHR$(18)
       1  SHIFT REVERSE =CHR$(146)
       1  CRSR↑ =CHR$(17)

2110 : 1  REVERSE =CHR$(18)
       1  SHIFT REVERSE =CHR$(146)

8000 : 1  SHIFT HOME =CHR$(147)
       6  CRSR↑ =CHR$(17)

8005 : 2  CRSR↑ =CHR$(17)
```

OTHELLO

Othello non è certamente un gioco nuovo, è basato sul REVERSI, gioco inventato nel 1888. Reversi si giocava su una tradizionale tavola da dama, usando pezzi bicolore, rossi da un lato, neri dall'altro. Nel suo splendido libro "Discowering old board games" (shire publications ltd., Aylesbury, 1980), R.C. Bell spiega che il nero comincia il gioco collocando un pezzo (con la faccia nera rivolta in alto), sopra uno dei 4 quadretti centrali della scacchiera. Il rosso replica collocando il suo primo pezzo, con la faccia rossa in alto, su un altro quadretto centrale. "Questi quattro quadretti vengono così coperti nei primi 4 turni di gioco, i giocatori continueranno poi, alternativamente, piazzando i loro pezzi su quadretti adiacenti ad uno occupato da un pezzo nemico". R.C. Bell scrisse.

Ogni pezzo nemico in linea (orizzontale, verticale o diagonale) tra l'ultimo piazzato ed un altro sempre del medesimo giocatore viene allora girato mostrando così il colore del giocatore.

Il vincitore è il giocatore che possiede il maggior numero di pezzi quando la scacchiera è completamente coperta, o quando né l'uno né l'altro possono muovere. Troverete il VIC lento a giocare, ma eccezionalmente attento, in questa versione del gioco. Potete scegliere se partire primi o secondi.

Per muovere inserirete le coordinate del quadretto dove volete andare, prima la distanza orizzontale e poi quella verticale (ascissa e ordinata), come se inseriresti un numero di due cifre (es. 34).

Remarks

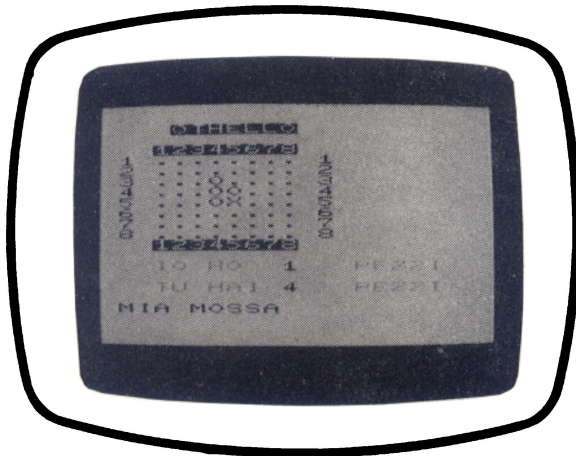
6	viene settato il bordo nero
10-110	prepara la scacchiera casella dopo casella inserendo il codice del punto. Le linee da 70 a 100 posizionano i primi 4 pezzi (si noti che la X è una variabile contenente il codice ascii della X, così la O contiene il codice ascii della O
120-170	viene data al giocatore la opportunità di muovere per primo
1000	comincia la routine cuore del programma, l'intelligenza del VIC "othellista"

1340 se H ed R sono uguali a zero allora si è nel caso in cui né il computer né il giocatore possono muovere, il gioco è finito.
In questo caso il controllo viene passato alla routine cominciante in 5000 che determina il vincitore

2000 comincia la routine che accetta la mossa del giocatore, questi inserirà zero se impossibilitato a muovere

3000 Comincia la routine che stampa e ristampa il bordo. Si noti l'uso del tasto HOME per cambiare il pezzo sotto ai vostri occhi

4000 comincia la routine che gira i pezzi

5000 fine del gioco

Suggerimenti

- cambiare la rappresentazione dei pezzi, ad esempio con dei cerchietti rossi e blu
- cambiare la strategia del computer cambiando i valori alle linee che vanno da 1220 a 1270, cambiandone due o tre potrete magari migliorare il gioco ma la risposta sarà più lenta
- aggiungere dei suoni quando i pezzi vengono girati

```
 1 REM OTHELLO
 4 PRINT"♥"
 5 X=ASC("X"):O=ASC("O")
 6 POKE36879,24
 10 DIMA(10,10)
 20 FORB=1TO10
 30 FORC=1TO10
 40 IFB<>1ANDC<>1ANDB<>10ANDC<>10THENA(
B,C)=ASC(".")
 50 NEXTC
 60 NEXTB
 70 A(5,5)=X
 80 A(6,6)=X
 90 A(6,5)=O
 100 A(5,6)=O
 110 P=0
 120 PRINTCHR$(28)"♥VUOI COMINCIARE TU"
 130 PRINT"      (1=SI 2=NO)
 140 INPUTW
 150 PRINT"♥S"
 160 GOSUB3000
 170 IFW=1THEN2000
 1000 PRINT" MIA MOSSA       "
 1010 S=0
 1020 T=X
 1030 H=0
 1040 FORA=2TO9
 1050 FORB=2TO9
 1060 IFA(A,B)<>46THEN1320
 1070 Q=0
```

```
1080 FORC=-1TO1
1090 FORD=-1TO1
1100 K=0
1110 F=A
1120 G=B
1130 IFA(F+C,G+D)<>STHEN1180
1140 K=K+1
1150 F=F+C
1160 G=G+D
1170 GOTO1130
1180 IFA(F+C,G+D)<>TTHEN1200
1190 Q=Q+K
1200 NEXTD
1210 NEXTC
1220 IFA=2 OR A=9 THEN Q=Q*2
1230 IFB=2 OR B=9 THEN Q=Q*2
1240 IFA=3 OR A=8 THEN Q=Q/2
1250 IFB=3 OR B=8 THEN Q=Q/2
1260 IF(A=2 OR A=9)AND(B=3 OR B=8)THEN
Q=Q/2
1270 IF(A=3 OR A=8)AND(Z=2 OR Z=9)THEN
Q=Q/2
1280 IFQ<H OR Q=0 OR (RND(1)>0.3ANDQ=H
)THEN1320
1290 H=Q
1300 M=A
1310 N=B
1320 NEXTB
1330 NEXTA
1340 IFH=0ANDR=0THEN5000
1350 IFH=0THEN1370
1360 GOSUB4000
1370 GOSUB3000
2000 INPUT"TUA MOSSA";R
2010 S=X
2020 T=O
2040 IFR=0THEN2090
2050 IFR<11ORR>88THEN2000
2060 M=INT(R/10)+1
2070 N=R-10*INT(R/10)+1
2080 GOSUB4000
2090 GOSUB3000
2100 GOTO1000
```

```
3000 PRINT"█"
3010 C=0
3020 H=0
3030 PRINT:PRINT"█      █OTHELLO█":PRINT
3035 PRINTCHR$(28)"     █12345678██"
3040 FORB=2TO9
3050 PRINTB-1;
3060 FORD=2TO9
3070 PRINTCHR$(A(B,D));
3080 IFA(B,D)=XTHENC=C+1
3090 IFA(B,D)=OTHENH=H+1
3100 NEXTD
3110 PRINTB-1
3120 NEXTB
3130 PRINTCHR$(28)"     █12345678██"
3140 PRINT
3150 PRINT"    IO HO ";CHR$(28)C"█  PEZ
ZI"
3152 PRINT
3155 PRINT"    TU HAI";CHR$(28)H"█  PEZ
ZI"
3157 PRINT
3170 RETURN
4000 FORC=-1TO1
4010 FORD=-1TO1
4020 F=M
4030 G=N
4040 IFA(F+C,G+D)<>STHEN4080
4050 F=F+C
4060 G=G+D
4070 GOTO4040
4080 IFA(F+C,G+D)<>TTHEN4140
4090 A(F,G)=T
4100 IF M=F AND N=G THEN4140
4110 F=F-C
```

```
4120 G=G-D
4130 GOTO4090
4140 NEXTD
4150 NEXTC
4160 RETURN
5000 IFC>HTHENPRINT" HO VINTO";C;H
5010 IFH<CTHENPRINT"  HAI VINTO";C;H
5020 IFH=CTHENPRINT"  PARITA'!"
5030 END
```

Lista simboli grafici

```
   4 : 1  SHIFT HOME =CHR$(147)

 120 : 1  SHIFT HOME =CHR$(147)

 150 : 1  SHIFT HOME =CHR$(147)

3000 : 1  HOME =CHR$(19)

3030 : 1  REVERSE =CHR$(18)
       1  SHIFT REVERSE =CHR$(146)

3035 : 1  REVERSE =CHR$(18)
       1  SHIFT REVERSE =CHR$(146)

3130 : 1  REVERSE =CHR$(18)
       1  SHIFT REVERSE =CHR$(146)
```

SINFONIA PER UN COMPUTER MALINCONICO

Questo, da cui prende titolo il libro, è un programma capace di comporre della musica che, come dice il titolo, è malinconica. Dopo aver udito la musica prodotta dal VIC troverete naturale il titolo che abbiamo voluto dare al programma. La principale struttura della musica viene dalle variabili A,B,C,D,E ed F che vengono ridefinite ad ogni RUN. Per come è scritto il programma gli accordi variano in riferimento al valore della variabile E, tali accordi si sovrappongono l'un l'altro dando origine a brusche pause e a bruschi inizi.

```
10 REM SINFONIA PER
15 REM UN MALINCONICO
16 REM COMPUTER
18 GOSUB 1000
20 A=128+INT(RND(1)*128)
30 B=128+INT(RND(1)*128)
40 C=128+INT(RND(1)*128)
50 D=128+INT(RND(1)*128)
60 E=INT(RND(1)*2)+1
70 F=2*E
75 POKE36878,F-1
90 FORH=1TO5*E
100 POKE36874,A
110 NEXTH
120 FORH=1TO5*E
130 POKE36875,B
140 NEXTH
150 FORH=1TO5*E
160 POKE36876,C
180 FORH=1TO5*E
190 POKE36877,D
200 NEXTH
210 FORH=1TO2*E
220 POKE36874,A
```

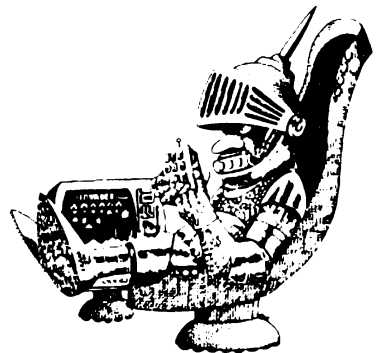
```
230 NEXTH
240 FORH=1TO5*E
250 POKE36876,C
260 NEXTH
500 RUN
1000 PRINT"□":REM CANCELLA
1010 POKE36879,16*(INT(RND(1)*16))+(IN
T(RND(1)*8))+8
1020 RETURN
```

Lista simboli grafici

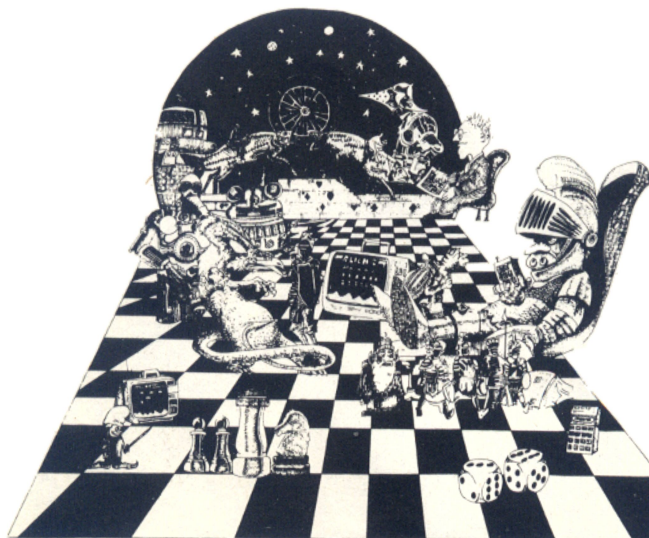
```
1000 : 1  SHIFT HOME =CHR$(147)
```

SINFONIA PER UN COMPUTER MALINCONICO CON UNO SHOW ZEN

Questo programma è lo stesso del precedente con l'aggiunta di una routine grafica.

```
10 REM SINFONIA PER
15 REM UN MALINCONICO
16 REM COMPUTER
17 REM CON UNO SHOW ZEN
18 GOSUB 1000
20 A=128+INT(RND(1)*128)
30 B=128+INT(RND(1)*128)
40 C=128+INT(RND(1)*128)
50 D=128+INT(RND(1)*128)
60 E=INT(RND(1)*2)+1
70 F=2*E
75 POKE36878,F-1
90 FORH=1TO5*E
100 POKE36874,A
110 NEXTH
120 FORH=1TO5*E
130 POKE36875,B
140 NEXTH
150 FORH=1TO5*E
160 POKE36876,C
180 FORH=1TO5*E
190 POKE36877,D
200 NEXTH
210 FORH=1TO2*E
220 POKE36874,A
230 NEXTH
240 FORH=1TO5*E
250 POKE36876,C
260 NEXTH
500 RUN
```

```
1000 PRINT"□":REM CANCELLA
1010 POKE36879,16*(INT(RND(1)*16))+(IN
T(RND(1)*8))+8
1020 FORJ=1TO5
1030 PRINT
1040 NEXT
1050 PRINT"■   SINFONIA PER UN      MAL
INCONICO COMPUTER "
1060 PRINT
1065 FORJ=1TORND(1)*20:PRINT" ";:NEXT
1070 PRINTCHR$(119+INT(RND(1)*15))
1075 FORJ=1TORND(1)*20:PRINT" ";:NEXT
1080 PRINTCHR$(119+INT(RND(1)*15))
1085 FORJ=1TORND(1)*20:PRINT" ";:NEXT
1090 PRINTCHR$(119+INT(RND(1)*15))
1095 FORJ=1TORND(1)*20:PRINT" ";:NEXT
1100 PRINTCHR$(119+INT(RND(1)*15))
1110 RETURN
```

Lista simboli grafici

```
1000 : 1  SHIFT HOME =CHR$(147)

1050 : 1  REVERSE =CHR$(18)
```

VIC NIM

Il VIC stampa un numero dispari di triangolini, vi chiede quanti volere prenderne, dopodichè anch'egli ne prende qualcuno, questo continua fino a che ne rimane uno solo, chi lo prende ha perso.

Remarks

20 determina il numero dei triangoli

50 determina il massimo numero di triangoli che può essere preso

100-130 stampa i triangoli

210-215 determina quanti triangoli prende il VIC, i random della linea 210 assicurano che il VIC faccia delle prese sbagliate di tanto in tanto. La linea 215 controlla invece che l'errore non sia così grosso (nel caso ci fosse) da infrangere le regole

280-290 cambia colore ad ogni manches del gioco

```
10 REM VIC NIM
20 M=0:E=0:Z=16+INT(RND(1)*8)
30 PRINT"■"
40 IF2*INT(Z/2)=ZTHENZ=Z+1
50 H=3+INT(RND(1)*2)
60 PRINT"■MASSIMA PRESA"H
70 GOSUB280
80 IFE>0THENPRINT"■NE HAI PRESI"E,"▒IL
VIC"Q
90 PRINT"■"
100 PRINT"■":FORK=1TOZ
110 PRINTK;"◤";
120 IFRND(1)>0.6THENPRINT
130 NEXTK
140 GOSUB280
145 PRINT"■QUANTI NE PRENDI";
150 INPUTE
160 IFE>HTHEN150
170 Z=Z-E
180 GOSUB280
190 IFZ<1THENPRINT"▒  HAI PRESO L'ULTI
MO "CHR$(28)"▓■■■IL VIC HA VINTO!!":EN
D
210 Q=Z-1-INT((Z-1)/(H+1))*(H+1)-INT(R
ND(1)*2)+INT(RND(1)*2)
215 IFQ<1ORQ>HTHEN210
220 GOSUB280
230 Z=Z-Q
240 IFZ=0THENPRINT"▒ IL VIC NE PRENDE■
"Q,,,"■■COSI' TU"CHR$(28)"▓■VINCI!!":E
ND
270 GOTO60
280 PRINT
285 PRINT"■##◣###▒##▓###■##◣###■##▒###
##"
290 RETURN
```

Lista simboli grafici

```
 30 : 1   SHIFT HOME =CHR$(147)

190 : 1   REVERSE =CHR$(18)
      4   CRSR← =CHR$(29)

240 : 2   CRSR← =CHR$(29)
      1   REVERSE =CHR$(18)
      1   CRSR← =CHR$(29)
```

EVOLUZIONE

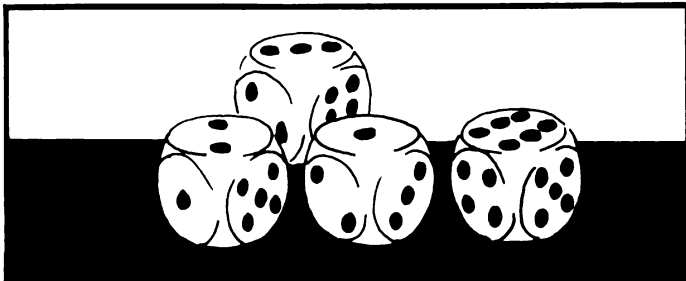
Nel 1970 John Conway, allora docente alla Cambridge University, inventò il GIOCO della VITA, che rapidamente divenne un best seller per computer, nell'ottobre del 70 fu descritto sulla nota "Scientific American".

Il gioco che simula la nascita, la crescita e la morte di una colonia di cellule, produce splendidi effetti visivi. Le regole date da Conway sono semplici, ma producono risultati veramente interessanti.

Le cellule nascono, vivono e muoiono in accordo coi dettami forniti dal DIO Conway. In accordo con le sue leggi, ciascuna cellula ne ha 8 contigue; ogni cellula che ne ha 2 o 3 vicine sopravvive fino alla prossima generazione, se ci sono 3, e solo 3, cellule intorno ad una, una nuova si genera, ed ogni cellula che ne ha 4 o più vicine muore per la sovrappopolazione.

Evoluzione è una versione del gioco della vita scritta appositamente per il VIC. Viene settato uno schermo nero con un bordo rosso.

Le cellule vengono disposte in un quadro 14 per 14, questo in accordo con le leggi di Conway. Il programma gira senza generare alcun suono, ma se vi fa piacere potrete aggiungere un suono con le seguenti linee:

```
6    POKE 36878,7
255  POKE 36876,240:POKE 36876,0
1075 POKE 36876,235:POKE 36876,0
```

Remarks

10 dimensiona le matrici che conterranno le cellule

20-80	genera la prima evoluzione della colonia, creando (linea 50) una cellula se il numero casuale è maggiore di 0,5. Gli elementi delle due matrici vengono posti uguali alla linea 60
90	il controllo viene passato alla subroutine che stampa la colonia
95	incrementa il numero dell'evoluzione
100-200	conta il numero di cellule intorno ad ogni cellula della colonia
210-220	vengono modificati gli elementi della matrice B in accordo con i risultati dello scandaglio or ora eseguito. Si noti che la matrice A non viene modificata, cosicchè un cambiamento dovuto alla evoluzione non interferisce con quella corrente che si sta studiando
1000-1090	stampa la colonia leggendola dalla matrice A

Suggerimenti

- cambiare le dimensioni delle matrici
- stampare ogni cellula in un diverso colore (rnd)
- aggiungere una routine che prenda il posto delle linee 20-80, e che vi permetta di inserire una vostra colonia di partenza
- aggiungere (come parte della routine di stampa) una routine che conti il numero totale di cellule, e se questo è sceso al di sotto di un numero prefissato, ad esempio 5, dia nuovamente il RUN. Questo nuovo RUN selezionerà un nuovo colore o di bordo o di sfondo

```
5 REM EVOLUZIONE
7 PRINT"[illegible]"
8 PRINT"[illegible]"
9 POKE36879,10:REM BORDO ROSSO,SCHERMO
NERO
10 DIMA(14,14),B(14,14)
20 FORX=2TO13
30 FORY=2TO13
50 IF RND(1)>0.5 THENA(X,Y)=1
60 B(X,Y)=A(X,Y)
```

```
70 NEXTY
80 NEXTX
90 GOSUB1000
95 G=G+1
100 FORX=2TO13
110 FORY=2TO13
120 C=0
```

```
130 IF A(X-1,Y-1)=1THENC=C+1
140 IF A(X-1,Y)=1THENC=C+1
150 IF A(X-1,Y+1)=1THENC=C+1
160 IF A(X,Y-1)=1THENC=C+1
170 IF A(X,Y+1)=1THENC=C+1
180 IF A(X+1,Y-1)=1THENC=C+1
190 IF A(X+1,Y)=1THENC=C+1
200 IF A(X+1,Y+1)=1THENC=C+1
210 IF A(X,Y)=1ANDC<>2ANDC<>3THENB(X,Y
)=0
220 IF A(X,Y)=0ANDC=3THENB(X,Y)=1
230 NEXTY
240 NEXTX
250 GOTO90
999 END
1000 PRINT"■"
1003 PRINT:PRINT"  ■EVOLUZIONE >>";G;"
":REM BIANCO,GIALLO
1005 PRINT
```

```
1010 FORX=1TO14
1020 FORY=1TO14
1030 A(X,Y)=B(X,Y)
1040 IFA(X,Y)=1THENPRINT"*";
1050 IFA(X,Y)=0THENPRINT" ";
1060 NEXTY
1070 PRINT
1080 NEXTX
1090 RETURN
```

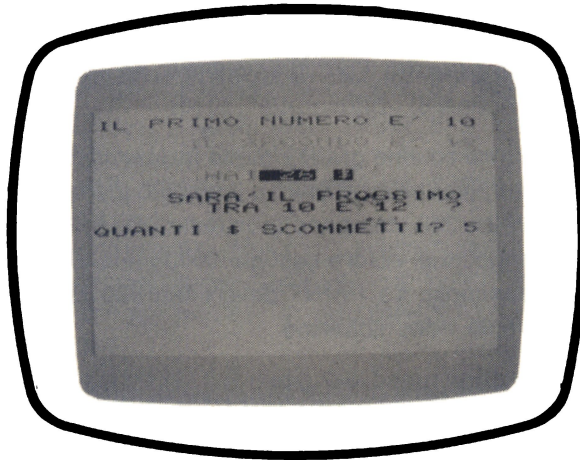
Lista simboli grafici

```
   7 : 1   SHIFT HOME =CHR$(147)

   8 : 1   REVERSE =CHR$(18)

1000 : 1   HOME =CHR$(19)
```

SCOMMETTIAMO

È questo un divertente programma, specie per i più giovani.

Il VIC sceglie due numeri tra l'uno ed il tredici, ti dice quanti $ hai a disposizione e ti chiede di scommettere sulla probabilità che il terzo numero (che sarà successivamente scelto) giaccia tra i due scelti. Se indovini, cioè il terzo numero è compreso fra i primi due, vinci il doppio di quello che hai puntato, se sbagli perdi invece quello che avevi scommesso. Finchè avrai soldi potrai continuare il gioco.

Parti con 20$ e puoi scommettere qualunque cifra purchè tu la possegga.

Remarks

30-40	genera i primi due numeri
50	verifica se i due numeri generati differiscono per più di una unità, se così non fosse il controllo è inviato alle linee 30-40 che genereranno due nuovi numeri

60 genera il terzo numero che in questa fase non è mostrato al giocatore

70 verifica che il terzo numero non sia uguale ad uno dei primi due, se lo fosse il controllo è rimandato alla linea 60 dove ne viene generato un altro

80-137 stampa il denaro in possesso del giocatore, i due numeri scelti e chiede al giocatore di scommettere

150-170 accetta le scommesse (150), chiede una nuova scommessa se il giocatore punta più di quanto possiede (160) e risponde di non barare se la scommessa è minore di un dollaro

180 attesa prima di rivelare il terzo numero

190-260 rivela il terzo numero (190), salta alla fine se la scommessa è minore di un $, salta alla routine di perdita se il numero non è compreso fra i primi due (210), si congratula col giocatore e gli aumenta il credito (220-30-40) se il numero era fra i primi due. Nelle linee 250-260 consola il giocatore e gli toglie dei dollari

270 verifica se il credito è più di un dollaro, altrimenti il controllo passa alla routine di "bancarotta" (310)

290-300 è questa una routine grafica molto interessante che potrete usare anche in altri programmi. La linea 290 stampa una linea e le linee da 292 a 298 stampano (con delle POKE) in posizioni casuali dei rettangolini di colore casuale. Dopo una breve pausa (299) il ciclo ricomincia. La PRINT della linea 290 fa sì che ci sia uno spostamento dello schermo

305 inizia un altro round del gioco

310-330 questa è la routine di fine gioco che stampa "il gioco è finito" e poi una serie infinita di scritte "tu sei rovinato"

Suggerimenti

- al posto dell'INPUT nella linea 150 usa una GET, con un contasecondi che costringa il giocatore a decidere velocemente e con una penalità di denaro se esce dal tempo

- aggiungi uno squillo di tromba per le vincite ed un rumoraccio per le perdite ed un super-rumoraccio per la "bancarotta"
- aggiungi qualche cinguettio (vedi manuale VIC) durante gli intervalli tra un round e l'altro
- fai cambiare casualmente l'ammontare della vincita e della perdita nei vari round, informando il giocatore di quanto vale una vincita o una perdita prima della scommessa

```
 10 REM SCOMMETTIAMO
 15 REM RIVISTO DA COMINI R.
 20 D=20
 25 PRINT"[CLR]"
 30 A=INT(RND(1)*13)+1
 40 B=INT(RND(1)*13)+1
 50 IFABS(B-A)<2THEN30
 60 C=INT(RND(1)*13)+1
 70 IFA=CORB=CTHEN60
 75 IFA>BTHENS=A:A=B:B=S
 80 PRINT"[DOWN]IL PRIMO NUMERO E'";A
 90 PRINT"      [DOWN]IL SECONDO E'";B
 100 PRINT
 120 PRINT"    [ ]    HAI[  ]";D;"$[ ]"
 125 PRINT
 130 PRINT"[    ]SARA'IL PROSSIMO"
 137 PRINT"[      ]TRA";A;"E";B"  ?"
 138 PRINT"[ ]QUANTI $ SCOMMETTI";
 150 INPUTE
 160 IFE>DTHEN150
 170 IFE<1THENPRINTCHR$(28)"     NON BAR
ARE  !!"
 180 FORZ=1TO999:NEXTZ
 190 PRINT"  [ ]IL MIO NUMERO E'";C
 200 IFE<1THEN280
 210 IFNOT(C>A AND C<B OR C<A AND C>B)T
HEN250
 220 PRINT"      MOLTO BENE"
 225 PRINTCHR$(28)"[    ]HAI VINTO $[ ]";2
*E"[ ]"
 230 D=D+2*E
 240 GOTO280
```

```
250 PRINT"      ■■SPIACENTE■■■■■■■■■■■■
■■HAI PERSO $";E;"■"
260 D=D-E
270 IFD<1THEN310
280 FORZ=1TO2000:NEXTZ
290 FORZ=1TO24:PRINT
292 L=INT(RND(1)*500)+1
294 C=INT(RND(1)*8)+1
296 POKE7680+L,160
298 POKE38400+L,C
299 FORX=1TO50:NEXTX
300 NEXTZ
305 GOTO25
310 PRINT"  IL GIOCO E' ■■FINITO!!■"
320 PRINT"■ TU SEI ROVINATO ";
325 FORZ=1TO999:NEXTZ
330 GOTO320
```

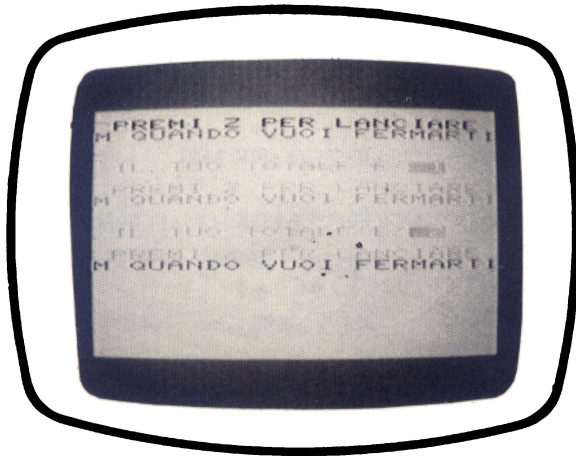
Lista simboli grafici

```
 25 : 1  SHIFT HOME =CHR$(147)

120 : 1  REVERSE =CHR$(18)
      1  SHIFT REVERSE =CHR$(146)

130 : 4  CRSR← =CHR$(29)

137 : 6  CRSR← =CHR$(29)

138 : 1  CRSR↑ =CHR$(17)

225 : 1  REVERSE =CHR$(18)
      4  CRSR← =CHR$(29)
      1  SHIFT REVERSE =CHR$(146)

250 : 1  REVERSE =CHR$(18)
      2  CRSR↑ =CHR$(17)
      11  SHIFT CRSR← =CHR$(157)
      1  SHIFT REVERSE =CHR$(146)

310 : 1  REVERSE =CHR$(18)
      1  SHIFT REVERSE =CHR$(146)
```

VIC—DADI

In questo programma voi ed il VIC (con sottofondo di tuoni, lampi e via discorrendo) lanciate a turno un dado cercando di realizzare un totale (con più lanci ovviamente) il più vicino possibile o addirittura uguale a 21, senza però superare il 21.

Questo gioco è una versione, utilizzante dadi, di Blackjack.

Giocare è molto semplice, dato il RUN non vi resta che muovere: premenzo "Z" lanciate i dadi e con "M" vi fermate, cioè vi tenete il punteggio totale che fino a quel punto avete raggiunto. Il vostro totale apparirà sullo schermo e potrete decidere se lanciare ancora o no. Una volta che avrete deciso di "stare" con quello che avete il VIC lancerà i suoi dadi e con due o tre lanci, in media, vi batterà.

Remarks

10	inizializza il contatore dei punti "umani" e quello dei punti del computer
20	il controllo è passato alla subroutine pazzerellona che casualmente sceglie suoni e colori

30-80	rileva ed esegue le mosse del giocatore
90-130	il computer decide se lanciare ancora o "stare"
140-250	si decide chi ha vinto e comincia una nuova partita
260	ciclo di ritardo chiamato più volte nel corso del programma
990	ad A$ viene assegnata una striga con i controlli colore
1000 in poi	da A$ viene prelevato un carattere per cambiare colore, viene poi generato un eccitante miscuglio di suoni e colori

```
 1 REM VIC-DADI
 7 PRINT"█":CANCELLA SCHERMO
 10 H=0:C=0
 20 GOSUB990
 30 PRINTB$;"█PREMI Z PER LANCIARE":GOS
UB1000:PRINTB$;"M QUANDO VUOI FERMARTI"
 35 GETM$:IFM$=""THEN35PRINTB$
 40 IFM$="M"THEN90
 50 H=H+INT(RND(1)*6)+1
 60 GOSUB260:GOSUB1000
 70 PRINT:PRINTB$;" IL TUO TOTALE E'█"H
"█"
 75 GOSUB260:GOSUB260:PRINT
 80 GOTO30
 90 IFC>HANDC<22ORC>21ORH>21ORH=21ANDC=
21THEN140
 100 C=C+INT(RND(1)*6)+1
 110 GOSUB260
 115 PRINT:PRINT
 120 GOSUB1000:PRINTB$;"    IL MIO TOTAL
E E'█"C"█"
 130 GOTO90
 140 GOSUB1000:PRINTB$;
 150 GOSUB260:GOSUB1000:GOSUB260
 170 IFH=COR(H>21ANDC>21)THEN240
 175 PRINT:PRINT:PRINTB$;
 180 IF(C>HORH>21)ANDC<22THENPRINT,"████
███HO";
 190 IF(C<HORC>21)ANDH<22THENPRINT,"████
███HAI";
```

```
 200 GOSUB1000:PRINTB$;" VINTO!!"
 210 GOSUB260
 220 GOSUB1000
 225 FORJ=1TO7:GOSUB260:NEXT
 226 GOSUB1000
 230 PRINT"]":RUN
 240 GOSUB1000:PRINTB$;"SIAMO PARI AMIC
O!"
 250 GOTO225
 260 FORE=1TO500:NEXT
 270 RETURN
 990 A$="[illegible]"
 1000 B$=MID$(A$,(INT(RND(1)*7)+1),1)
 1010 FORJ=1TOINT(RND(1)*10)+1
 1020 POKE36878,INT(RND(1)*10)+6
 1030 FORK=1TOINT(RND(1)*7)+2
 1040 POKE36876,K+180-INT(RND(1)*30)
 1050 NEXTK,J
 1070 POKE36879,INT(RND(1)*3)+28:POKE36
879,8
 1075 POKE36879,INT(RND(1)*3)+28
 1080 POKE36879,INT(RND(1)*3)+28
 1090 POKE36878,0
 1100 RETURN
```

Lista simboli grafici

```
   7 : 1  SHIFT HOME =CHR$(147)

  30 : 1  CRSR← =CHR$(29)

  70 : 1  REVERSE =CHR$(18)
       1  SHIFT REVERSE =CHR$(146)

 120 : 1  REVERSE =CHR$(18)
       1  SHIFT REVERSE =CHR$(146)

 180 : 5  SHIFT CRSR← =CHR$(157)

 190 : 5  SHIFT CRSR← =CHR$(157)

 230 : 1  SHIFT HOME =CHR$(147)
```

GOLF

In questo gioco vi trovate in un campo da golf a 9 buche. La distanza di ogni buca varia ogni volta che giocate.

Il programma fa uso dell'istruzione POKE per disegnare il campo e muovere la palla. Il suono è veramente interessante quando la palla va in buca. Non è necessario spiegare qui il funzionamento del programma perchè questo compito è affidato alle molte REM che in esso compaiono.

```
 10 REM GOLF
 15 POKE36879,28
 20 SC=0:REM PUNTEGGIO
 30 AV=0:REM MEDIA
 100 FORA=1TO9:REM CONTEGGIA IL NUMERO
DELLA BUCA
 105 T=0:REM TIRI PER LA BUCA
 110 D=INT(RND(1)*7)+14:REM
 120 PRINT"⊐"
 125 C=0
 130 M=0
```

```
400 GOSUB6000
500 GOSUB5000
510 IFM<>DTHEN500
520 IFABS(M-D)<2THEN8000
530 GOTO500
540 END
5000 PRINT"███BUCA NUMERO"A
5005 PRINT,D*10"METRI"
5010 PRINT
5012 PRINT" █PUNTI"SC"█  TIRO"T"  "
5015 IFD=MTHEN8000
5020 PRINT"█DISTANZA:";
5025 IFD-M=>10THENPRINT"█"(D-M)*10"MET
RI    "
5026 IFD-M<10THENPRINT"  "(D-M)*10"METR
I            "
5030 PRINT"█"
5040 INPUT"████CHE MAZZA";C
5042 T=T+1
5044 GOSUB7000
5045 POKE38730+M,1
5047 IFM>DTHENC=-C
5060 M=INT(M+C/3*RND(1+1))
5065 C=0
5070 POKE8010+M,81
5080 POKE38730+M,2
5090 RETURN
5099 END
6000 REM DISEGNA IL PRATO
6010 FORQ=8032TO8119
6020 POKEQ,102
6030 POKEQ+38752-8032,5
6040 NEXT Q
6050 POKE8032+D,74:POKE8033+D,75
6060 POKE38752+D,2:POKE38753+D,2
6065 POKE8010+M,81
6070 POKE38730+M,2
6100 RETURN
7000 FORW=128TO180+3*M
7005 POKE36878,15
7020 POKE36875,W
7030 POKE36874,W
7040 NEXTW
```

```
 7050 POKE36875,0
 7060 POKE36874,0
 7500 RETURN
 8000 T=T-1
 8002 POKE36876,220
 8003 FORP=1TO5:NEXT
 8004 POKE36876,0:POKE36876,225:POKE368
76,0
 8010 PRINT"██SEI ARRIVATO IN"T
 8020 POKE8010+M,1
 8030 POKE8032+M,81
 8040 POKE38774+M,2
 8050 POKE8033+M,102
 8060 POKE38775+M,5
 8062 SC=SC+T
 8065 PRINT"███████MEDIA"INT(SC/A+.5)
 8070 FORH=1TO5000:NEXT
 8990 NEXTA
 9000 END
```

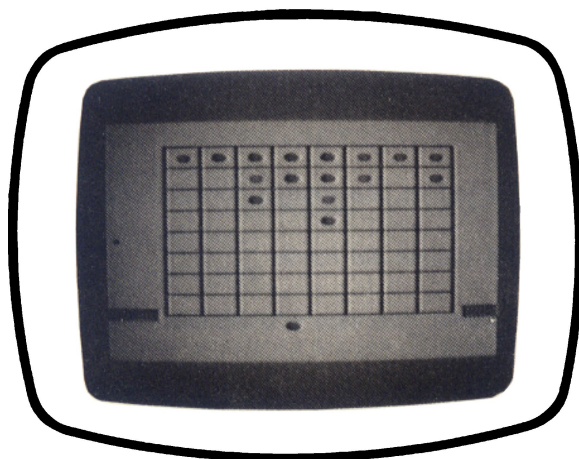
Lista simboli grafici

```
  120 : 1  SHIFT HOME =CHR$(147)

 5000 : 1  HOME =CHR$(19)
        1  REVERSE =CHR$(18)

 5020 : 1  CRSR← =CHR$(29)

 5040 : 4  CRSR← =CHR$(29)

 8010 : 2  CRSR← =CHR$(29)

 8065 : 7  CRSR← =CHR$(29)
```

FILETTO GRAVITAZIONALE

(DI G. DI BELLA)

Il gioco è molto semplice e divertente. Due giocatori, a turno e contrassegnati da colori diversi devono cercare di depositare su una scacchiera 8X8 cinque pallini consecutivi in file orizzontali, verticali, diagonali. Direte: "ma ci abbiamo già giocato un sacco di volte (a scuola)!" Non si tratta di disegnare, come al solito, i pallini su di un normale foglio quadrettato, ma, ecco la novità, di sganciarli impilando ogni nuovo pallino sul vecchio, nelle diverse colonne della scacchiera.

In altre parole, osservando la figura è possibile depositare il pallino A nella posizione (6,1) solo se sono già presenti i pallini B (7,1) e C (8,1). Il calcolatore si limita quindi a riconoscere, mediante la scansione completa della scacchiera, se qualcuno ha realizzato il filetto. Il gioco gira sul VIC-20 nella sua configurazione base e, anche se il programma centrale in sè occupa ben poco spazio, le stringhe per la creazione della scacchiera, il corposo gruppo di variabili, la piccola presentazione e qualche altro gadget lasciano liberi in esecuzione poco più di 150 BYTE. Il listato va copiato con molta attenzione, senza omettere nulla, in particolare il REMARK della linea 5, visto che i suoi caratteri vengono poi utilizzati nella creazione dell'intestazione.

Remarks

5-28	La "macchinosa intestazione del programma si avvale anche della subroutine 2000; notare quanto occorre per generare due secondi di presentazione. Seguono le istruzioni di gioco
30-200	input nomi e stampa del campo di gioco. Il contatore dei pallini depositati viene posto a zero
210-220	parte per primo il giocatore col pallino porpora
230-305	si preleva e si smista la mossa del giocatore
310-330	impilazione del pallino
340-360	dalla posizione video del pallino or ora depositato si risale alle coordinate X, Y per aggiornare la situazione della scacchiera (matrice A); viene quindi incrementato il N° di pallini depositati e, se questi sono minori di 9 si salta tutta la fase di scansione della scacchiera
400-665	é il programma vero e proprio, la ricerca dei cinque pallini in fila, dapprima sulle file, poi sulle colonne ed infine su tutte le diagonali con almeno 5 caselle
680	se la scacchiera si è riempita completamente e nessuno ha realizzato il filetto la partita si chiude in parità
800-810	il pallino è stato depositato senza provocare nulla, tocca all'altro giocatore
900	suono che indica una mossa non corretta
850	tonfo della pallina sul fondo
910	pulisce il buffer riservato alla tastiera dagli ultimi codici dei caratteri premuti
920-1090	proclamazione visiva e sonora del vincitore con la richiesta per un'eventuale nuova partita

```
 5 REM FILETTO GRAVITAZIONALE
 7 PRINT"":P=7:C=5:A=4110:FL=30720:I=7
702:POKE36879,29
 10 POKEI,81:POKEI+FL,C:FORS=1TO14:M=2+
S:GOSUB2000:NEXT:I=7702
 15 C=4:P=5:A=4102:POKEI,81:POKEI+FL,C:
FORS=1TO7:M=5+S:GOSUB2000:NEXT
 20 PRINT"            DI G. DI BE
LLA"
21 FORI=1TO3000:NEXT
22 PRINT"     METTI 5 PALLINI IN FILA
: IN DIAGONALE, ORIZZONTALE O VERTICALE
"
23 PRINT"CON  SI VA A SINISTRA C
ON  A DESTRA CON  SI 'SGA
NCIA'"
27 PRINT"      PREMI UN TASTO"
28 IFPEEK(197)=64THEN28
30 CLR:GOSUB910:PRINT"":POKE3687
9,31
40 INPUT" 1°NOME";P1$
41 INPUT" 2°NOME";P2$
50 A$=" ┌─┬─┬─┬─┬─┬─┬─┐"
55 B$="    ├─┼─┼─┼─┼─┼─┼─┤"
56 C$="    │ │ │ │ │ │ │ │ │"
60 PRINT""CHR$(28)"▒▒"A$"▒▒▒";
70 FORI=0TO6:PRINTC$:PRINTB$:NEXT:PRIN
TC$
80 PRINT" └─┴─┴─┴─┴─┴─┴─┘"
90 PRINT"‾‾  ‾‾‾"
200 V=36878:G1=36874:SC=36879:D=0
210 X=7746:PL=1:CL=4:S=44
220 POKEX,81:POKEX+30720,CL
225 FORW=0TO20:NEXT
230 M=PEEK(197):IFM=64THEN230
240 IFM=41THEN280
250 IFM=42THEN300
260 IFM<>18THEN230
265 IFX=7767THENGOSUB900:GOTO230
270 POKEX,32:X=X+1:GOTO220
280 IFX=7746THENGOSUB900:GOTO230
290 POKEX,32:X=X-1:GOTO220
```

```
300 IFPEEK(X+S)<>32THENGOSUB900:GOTO22
0
305 POKEX,32
310 S=S+44:IFS>352THEN330
320 IFPEEK(X+S)=32THENPOKEX+S-44,32:PO
KEX+S,81:POKEX+S+30720,CL:GOTO310
330 POKEX+S-44,81:POKEX+S-44+30720,CL:
GOSUB850
340 I=(S-44)/44
350 J=(X-7749)/2+1
355 GOSUB5000
360 A(I,J)=CL:D=D+1:IFD<9THEN800
400 C=0:FORK=1TO8
410 IFA(I,K)<>CLTHENC=0:GOTO430
420 C=C+1:P(C)=7749+I*44+(K-1)*2:IFC=5
THEN1000
430 NEXT
450 IFI>4THEN500
460 C=0:FORK=1TO8
470 IFA(K,J)<>CLTHENC=0:GOTO495
480 C=C+1:P(C)=7749+K*44+(J-1)*2:IFC=5
THEN1000
495 NEXT
500 FORH=5TO8:C=0:FORK=HTO1STEP-1:S=H-
K+1
510 IFA(S,K)<>CLTHENC=0:GOTO530
520 C=C+1:P(C)=7749+S*44+(K-1)*2:IFC=5
THENGOTO1000
530 NEXTK,H
550 FORH=2TO4:C=0:FORK=8TOHSTEP-1:S=H-
K+8
560 IFA(S,K)<>CLTHENC=0:GOTO580
570 C=C+1:P(C)=7749+S*44+(K-1)*2:IFC=5
THENGOTO1000
580 NEXTK,H
600 FORH=4TO1STEP-1:C=0:FORK=HTO8:S=K-
H+1
610 IFA(S,K)<>CLTHENC=0:GOTO630
620 C=C+1:P(C)=7749+S*44+(K-1)*2:IFC=5
THENGOTO1000
630 NEXTK,H
650 FORH=4TO2STEP-1:C=0:FORK=1TO9-H:S=
H+K-1
```

```
 655 IFA(S,K)<>CLTHENC=0:GOTO665
 660 C=C+1:P(C)=7749+S*44+(K-1)*2:IFC=5
THENGOTO1000
 665 NEXTK,H
 680 IFD=64THENPRINT"███PARITA'█":GOTO1
046
 800 IFPL=1THENX=7767:CL=5:PL=2:S=44:GO
TO220
 810 GOTO210
 850 POKEV,15:POKEG1+2,170:FORW=0TO7:NE
XT:POKEG1+2,0:POKEV,0:RETURN
 900 POKEV,15:FORI=255TO128STEP-1:POKEG
1,I:NEXT:POKEG1,0:POKEV,0:RETURN
 910 FORI=0TO9:GETP1$:NEXT:RETURN
 920 POKEV,15:POKEG1+2,L+128:L=L+1:RETU
RN
 1000 PRINT"███BRAVO "CHR$(28);:IFPL=1T
HENPRINTP1$;:GOTO1020
 1010 PRINTP2$;
 1020 PRINT"█ HAI VINTO!██":GOSUB910
 1025 FORJ=0TO8
 1030 FORI=1TO5:POKEP(I),32:GOSUB920:NE
XT
 1035 FORW=0TO150:NEXT
 1040 FORI=1TO5:POKEP(I),81:GOSUB920:NE
XT
 1045 FORW=0TO450:NEXTW,J:FORW=LTO254:P
OKEG1+2,W:NEXT
 1046 FORI=WTO0STEP-1:POKEG1+2,I:NEXT:P
RINT"██PER CONTINUARE PREMI C":POKEV,0:
POKEGI+2,0
 1050 GETP1$:IFP1$=""THEN1050
 1060 IFP1$="C"THEN30
 1070 POKE36881,159
 1080 PRINT"███████████";CHR$(28);"GAMEO
VER":POKEV,15
 1090 FORI=159TO39STEP-1:POKE36881,I:PO
KEG1+2,I+95:NEXT:POKEV,0:END
 2000 FORI=7702TO7702+M:POKEI-1,32:POKE
I,81:POKEI+FL,C:FORT=0TO20:NEXTT,I:POKE
I-1,32
 2010 FORW=ITOI+P*22STEP22:POKEW-22,32:
POKEW,81:POKEW+FL,C:NEXT:POKEW-22,32
```

```
 2020 POKEW,PEEK(A+S)+64:POKEW+FL,2:RET
URN
 5000 FORTK=38TO42:POKE36865,TK:NEXTTK
 5010 FORTK=42TO38STEP-1:POKE36865,TK:N
EXTTK:RETURN
```

Lista simboli grafici

```
  7 : 1   SHIFT HOME =CHR$(147)

 20 : 13   CRSR↑ =CHR$(17)

 22 : 1   SHIFT HOME =CHR$(147)
      1   CRSR↑ =CHR$(17)

 23 : 2   CRSR↑ =CHR$(17)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
      1   CRSR↑ =CHR$(17)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
      1   CRSR↑ =CHR$(17)
      8   CRSR← =CHR$(29)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
      2   CRSR↑ =CHR$(17)

 27 : 1   CRSR↑ =CHR$(17)

 30 : 1   SHIFT HOME =CHR$(147)
      5   CRSR↑ =CHR$(17)

 41 : 1   CRSR↑ =CHR$(17)

 60 : 1   SHIFT HOME =CHR$(147)
      4   CRSR↑ =CHR$(17)

 80 : 2   CRSR← =CHR$(29)
      1   HOME =CHR$(19)
```

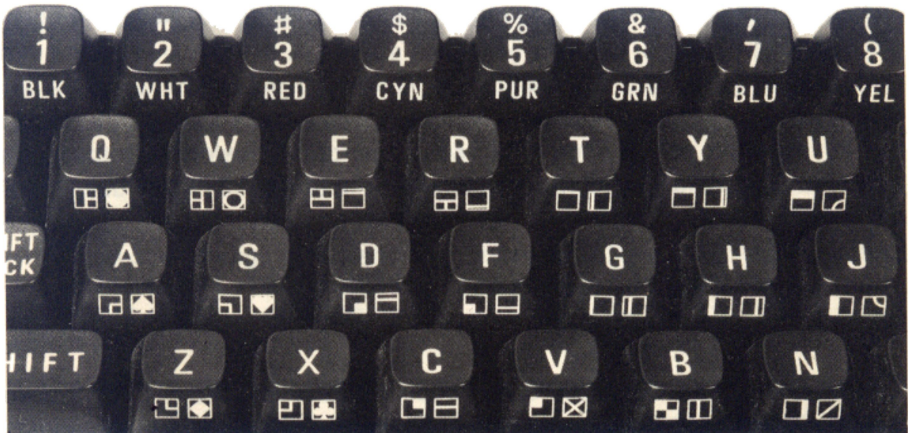
```
  90 : 1   HOME =CHR$(19)
       5   CRSR↑ =CHR$(17)
       17   CRSR← =CHR$(29)

 680 : 1   HOME =CHR$(19)
       1   REVERSE =CHR$(18)
       1   SHIFT REVERSE =CHR$(146)

1000 : 1   HOME =CHR$(19)
       1   REVERSE =CHR$(18)

1020 : 1   SHIFT REVERSE =CHR$(146)

1046 : 1   HOME =CHR$(19)
       1   CRSR↑ =CHR$(17)

1080 : 1   SHIFT HOME =CHR$(147)
       3   CRSR↑ =CHR$(17)
       6   CRSR← =CHR$(29)
```

PROVARIFLESSI

(di R. COMINI)

Questo, come dice il nome, non è un gioco, ma un provariflessi, uno strumento atto a verificare la nostra velocità di reazione sotto uno stimolo sia acustico che visivo.

Funzionamento

Una volta dato il RUN "gira" la presentazione. Per partire basterà premere un tasto qualunque. Tre segnali acustici indicheranno l'inizio della prova, dopodichè lo schermo si farà completamente bianco, dopo un lasso di tempo variabile (pochi secondi) tutto lo schermo diverrà rosso e si udrà in contemporanea un fischio. Nell'istante in cui udremo il fischio e lo schermo diverrà rosso dovremo, il più velocemente possibile, premere la barra. Al segnale acustico il calcolatore inizia un veloce conteggio che si ferma appunto con la pressione della barra. Dopo aver premuto la barra al centro comparirà il numero a cui siamo riusciti a bloccare il conteggio. Tale numero potrà

al minimo essere 1 (praticamente impossibile riuscirvi) ed al massimo 350, nel qual caso vuol dire che siamo arrivati troppo tardi a premere la barra: poichè questo è il limite massimo di tempo che abbiamo a disposizione per bloccare il conteggio dopo aver udito la segnalazione. Il numero da noi totalizzato rimane al centro dello schermo qualche istante poi lo schermo si fa di nuovo bianco e dopo un'attesa, che è variabile, come già detto, ci sarà la nuova segnalazione acustica cui rispondere (pressione della barra).

Dopo che il tutto si sarà ripetuto 5 volte (tante sono le prove a nostra disposizione) apparirà la classifica con il nostro livello di riflessi. Il punteggio da noi realizzato ci classificherà in uno dei 10 livelli che il calcolatore ha a disposizione. Il primo livello, difficilissimo da raggiungere, è "BOREALE", e l'ultimo, praticamente corrisponde a non aver mai premuto in tempo la barra, è "NON ESISTE". Tra il primo ed il decimo livello ci sono naturalmente i livelli intermedi. Alla fine della prova il computer ci chiederà se vogliamo farne un'altra, se rispondiamo sì tutto ricomincia, altrimenti il programma termina di "girare".

ATTENZIONE: se siamo troppo nervosi e premiamo la barra prima del segnale acustico l'intera prova ci verrà, alla fine, annullata e ci verrà segnalato il nostro nervosismo.

Con questo riflessometro possiamo quindi, più per divertimento che per altro, cimentarci col computer cercando di migliorare sempre più i nostri riflessi.

Il programma

Il programma è scritto per il 50% in BASIC e per il restante 50% in linguaggio macchina: l'uso del linguaggio macchina rende possibile una notevole velocità.

Le routine in linguaggio macchina si inseriscono nel seguente modo: per ogni routine è riportato in Figura 1 il contenuto di tutte le locazioni di memoria, ad esempio per la "controlla" sono riportati 17 valori numerici. Ora poichè la "controlla" va (come specificato dalla tabella di Fig. 1) dalla locazione 6335 alla 6351 dovremo introdurre i valori numerici dati tra queste due locazioni. Per fare questo creeremo una linea di programma così:

```
10000 FORI = 6335 TO 6351 : PRINTI : INPUTX : POKEI, X : NEXT
```

fatto questo si dà RUN 10000 e uno dopo l'altro si inserisono i valori della routine. Fatto questo la nostra routine è in memoria, e sempre con questo

procedimento (modificando di volta in volta la linea 10000) caricheremo in memoria tutte le altre routine. Attenzione: con lo stesso procedimento dovremo pure riportare i 506 valori numerici inerenti alla immagine della classifica, che vanno inseriti dalla locazione 6700 alla 7205 (Fig. 2). Fatte tutte queste operazioni il difficile è ormai superato. È necessario fare attenzione perchè un solo numero sbagliato può causare la perdita di tutto il programma (il computer si potrebbe "inchiodare"): digitare i vari numeri con calma è indispensabile.

Ora non resta che inserire il testo BASIC (ricordandosi prima di togliere la linea 10000 che abbiamo appena usato), prestando molta attenzione per gli indirizzi delle SYS. Completato il tutto è utile, prima di dare il RUN, salvare il tutto su cassetta, onde evitare di perdere tutto, magari per un errore banale.

Il Save

Se si desse a questo punto il SAVE, quello che salveremmo su cassetta o disco sarebbe solo il programma BASIC e non le routine in linguaggio macchina: questo avviene perchè il computer non "vede" queste routine, in quanto i puntatori riguardanti il programma puntano alla fine del BASIC. Il resto viene considerato memoria vuota a disposizione del sistema operativo (per le variabili ecc), quindi procederemo così: controlliamo innanzitutto il list onde verificare grossolani errori, dopodichè diamo un CLR.

Vediamo ora i contenuti delle locazioni 45 e 46 con

```
PRINT PEEK (45), PEEK (46).
```

I due valori ottenuti sono parte bassa e parte alta del puntatore alla fine del programma BASIC. Per avere l'indirizzo effettivo bisogna moltiplicare la parte alta per 256 e aggiungere la parte bassa. Scriviamo ora i due numeri su un foglio che faremo attenzione a non perdere e procediamo allo spostamento dei puntatori.

Diamo

```
POKE45,152 : POKE46,28 : POKE47,152 : POKE48,28
POKE49,152 : POKE50,28
```

Fatto? Ora il sistema operativo "vede" il programma come se finisse alla locazione 7320. Possiamo perciò dare il SAVE, e su cassetta verrà salvato

non solo il testo BASIC ma anche tutta la memoria con le routine in L.M.

Diamo pure il RUN senza paura (il tutto è salvo). Se ci fossero errori nel BASIC o volessimo apportare delle modifiche dovremmo, prima di toccare il list, riportare i tre puntatori (45 e 46 47 e 48 48 e 50) al valore iniziale che avevano (lo abbiamo segnato su un foglio)
POKE45XXX : POKE46YYY : ecc, dove XXX rappresenta la parte bassa ed YYY la parte alta del puntatore.

Fatte le modifiche prima di salvare il programma dovremo nuovamente vedere il valore dei puntatori (basterà quello del primo, loc. 45 e 46) e ricordarlo per eventuali altre modifiche. Sposteremo poi i puntatori, come già fatto, e salveremo. Tutte queste operazioni andranno fatte per ogni modifica.

Un consiglio: per non stare a scrivere ogni volta il valore dei puntatori è utile creare una linea cosi

```
REM PUNTATORI     000 – 000
```

e dopo aver visto i valori di questi, li scriveremo nella linea al posto degli zeri; Es. se i valori fossero 100,26 scriveremo cosi:

```
REM PUNTATORI     100 – 026
```

senza nè allungare nè accorciare la linea (cambierebbero i valori). In questo modo sapremo sempre come modificare i puntatori onde apportare modifiche.

Si sconsiglia vivamente di allungare il testo BASIC.

Suggerimenti

- una modifica semplice da apportare è quella relativa al colore delle frecce nella graduatoria finale: basta "pokare" nella locazione 6241 il valore del colore desiderato.

- per cambiare invece il colore di tutta l'immagine della graduatoria si agirà con POKE6204,XX : POKE6218,XX dove XX rappresenta il valore del colore desiderato.

Remarks

20	inizializza il volume, il colore e aggiusta l'inizio della routine CONTROLLA
100-105	inizializzazione della CELLA B e segnale acustico di inizio prova
110	ciclo che permette la ripetizione della prova per 5 volte
500	attesa random
520-560	lancio della routine CONTEGGIA, e visualizzazione del risultato
570	I risultati delle 5 prove vengono sommati (sono prelevati dalle 5 CELLE B)
580-585	viene fatta una pesata media delle prove ed il risultato è memorizzato in F1
587-589	segnalazione di barra premuta anzitempo ed annullamento della prova
590-600	visualizzazione risultato e classifica
610-640	conclusione
1000-1150	presentazione
2000-2050	musichetta iniziale

```
1 REM--------------------------------------
2 REM                                     *
3 REM         RIFLESSOMETRO               *
4 REM                                     *
5 REM              BY                     *
6 REM         COMINI RENATO               *
7 REM             1983                    *
8 REM--------------------------------------
9 GOTO2000
10 REM PUNTATORI                     110 022
20 POKE36878,15:POKE36879,25:POKE6335,
165:POKE6015,0
```

```
 100 SYS6175
 105 PRINT"":FORW=1TO3:POKE36876,230:F
ORG=1TO600:NEXTG:POKE36876,0:FORH=1TO50
:NEXTH,W
 110 FORJ=1TO5
 120 PRINT"":POKE36879,25
 500 FORL=1TOINT(RND(1)*300)+450:SYS633
5:NEXTL
 520 SYS6020
 530 Y=(PEEK(6001)*256)+PEEK(6000)
 540 PRINT""Y
 550 FORS=1TO500:NEXTS
 560 NEXTJ:POKE36879,25
 570 F1=0:FORI=6003TO6011STEP2:F1=F1+(P
EEK(I)*256+PEEK(I-1)):NEXT
 580 F1=INT(F1/175)+1
 585 IFF1=11THENF1=10
 586 POKE6800,1:POKE6754,1
 587 IFPEEK(6015)=0THEN590
 588 PRINTCHR$(159)"SEI TROP
PO NERVOSO"CHR$(28)"RIPRO
VA"CHR$(31)
 589 :FORI=1TO2500:NEXT:GOTO1000
 590 SYS6195:FORJ=1TO300:NEXT:FORI=1TO9
:FORK=1TO100:NEXTK:SYS6285:NEXTI
 600 FORI=1TOF1:SYS6285:FORQ=1TO200:NEX
TQ:NEXTI:FORJ=1TO2000:NEXTJ
 610 PRINT"GIOCHI ANCOR
A ?(S/N)"
 620 GETR$:IFR$=""THEN620
 625 IFR$<>"S"ANDR$<>"N"THEN620
 630 IFR$="S"THEN10
 640 IFR$="N"THENEND
 1000 PRINT"PROVA I TUOI"
:PRINT"RIFLESSI"
 1005 PRINT"PREMI U
N TASTOPER INIZIARE"
 1010 FORE=1TO5:FORI=4TO7:FORJ=38605TO3
8612:POKEJ,I:NEXTJ
 1020 FORK=1TO30:NEXTK,I,E
 1030 POKE6754,1:POKE6800,1
 1100 SYS6195:FORI=1TO300:NEXT
 1105 TI$="000000"
```

```
 1110 SYS6285:FORI=1TO80:NEXT:IFTI$>"00
0008"THEN1000
 1120 IFPEEK(197)=64THEN1110
 1150 GOTO10
 2000 DATA225,360,225,360,225,240,228,1
20,231,360,231,240,228,120,231,240,232,
120
 2010 DATA235,720,240,360,235,360,231,3
60,225,360,235,240,232,120,231,240,228,
120,225,480
 2015 X=36875:POKE36878,15:PRINT"▯":POK
E36879,25
 2020 FORI=1TO19:READN:READD:POKEX,N:GO
SUB2050:FORT=1TOD/2:NEXTT:POKEX,0:NEXT:
GOTO1000
 2050 H=INT(RND(1)*500):POKE7680+H,160:
POKE38400+H,INT(RND(1)*7+1):RETURN
```

Lista simboli grafici

```
 105 : 1   SHIFT HOME =CHR$(147)

 120 : 1   SHIFT HOME =CHR$(147)

 540 : 8   CRSR← =CHR$(29)
       10   CRSR↑ =CHR$(17)

 588 : 1   SHIFT HOME =CHR$(147)
       8   CRSR↑ =CHR$(17)
       2   CRSR← =CHR$(29)
       1   SHIFT REVERSE =CHR$(146)
       1   CRSR↑ =CHR$(17)
       12   SHIFT CRSR← =CHR$(157)

 610 : 1   SHIFT HOME =CHR$(147)
       10   CRSR↑ =CHR$(17)
       4   CRSR← =CHR$(29)
       1   CRSR↑ =CHR$(17)
       12   SHIFT CRSR← =CHR$(157)
```

```
1000 : 1   SHIFT HOME =CHR$(147)
       8   CRSR↑ =CHR$(17)
       12   CRSR← =CHR$(29)
       1   REVERSE =CHR$(18)
       1   SHIFT REVERSE =CHR$(146)

1005 : 1   HOME =CHR$(19)
       15   CRSR↑ =CHR$(17)
       4   CRSR← =CHR$(29)
       1   CRSR↑ =CHR$(17)
       13   SHIFT CRSR← =CHR$(157)

2015 : 1   SHIFT HOME =CHR$(147)
```

CODICE MACCHINA DELLE VARIE ROUTINE

I numeri vanno inseriti per colonne

- CONTEGGIA (6020 - 6143)

169	141	23	23
200	113	165	105
141	23	197	2
12	32	201	141
144	6	32	192
169	24	208	23
42	173	203	173
141	113	173	207
15	23	112	23
144	201	23	105
169	1	141	2
0	208	114	141
141	20	23	207
[illegible]	173	173	23
23	112	113	173
141	23	23	213
113	201	141	23
23	94	115	105

24	208	23	2
173	13	234	141
112	169	24	213
23	94	24	23
105	141	173	169
1	114	187	0
141	23	23	141
112	169	105	12
23	1	2	144
173	141	141	96
113	115	187	
23	23	23	
105	76	173	
0	216	192	

- DELAY (6150 - 6169)

162	234	239
0	192	96
160	8	
0	208	
200	246	
234	232	
234	224	
234	7	
234	208	

- PRELOCB (6175 - 6191)

169	169	96
114	115	

141	141
187	192
23	23
141	141
207	213
23	23

- FLASH (6195 - 6235)

162	157	133
0	255	1
189	30	96
44	169	
26	4	
157	157	
0	255	
30	150	
169	232	
4	224	
157	250	
0	208	
150	240	
232	169	
208	25	
242	133	
189	0	
43	169	
27	150	

- COL (6240 - 6279)

169	145	0	160	37
4	0	160	15	145
76	160	23	145	0
103	1	145	0	96
24	145	0	160	
169	0	160	36	
1	160	14	145	
160	22	145	0	
0	145	0	160	

- SPOSTA (6285 - 6329)

165	169	165
0	25	0
201	133	105
165	0	44
208	169	133
20	150	0

```
165        133        165
  1          1          1
201         76        105
151         96          0
208         24        133


 14         32          1
 32        101         76
101         24         96
 24         24         24
```

- CONTROLLA (6335 - 6351)

```
165        169
197         96
201        141
 64        191
240         24
 10         96
169
  1
141
127
 23
```

LE ROUTINE IN LINGUAGGIO MACCHINA

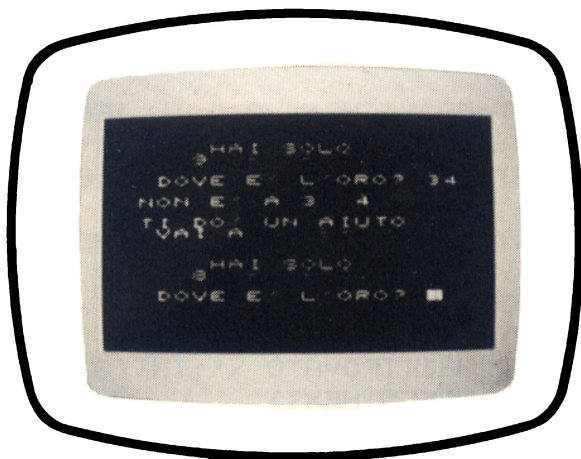
CONTEGGIA	DA 6020 A 6143
DELAY	DA 6150 A 6169
PRELOCB	DA 6175 A 6191
FLASH	DA 6195 A 6235
COL	DA 6240 A 6279
SPOSTA	DA 6285 A 6329
CONTROLLA	DA 6335 A 6351

FEBBRE DELL'ORO

È questo un semplicissimo programma che puoi usare come cuore di altri programmi quando vuoi nascondere qualcosa in una scacchiera.

In questo caso la scacchiera (o meglio il terreno di gioco) é 10 per 10 attraverso varie mosse cercherete di trovare l'oro.

Remarks

10-20	l'oro viene nascosto
30	ciclo che conta i tentativi
60	accetta la mossa del giocatore, tramite due numeri
130-190	esegue la mossa e fornisce precise indicazioni su come trovare l'oro
210-220	se l'oro non è stato trovato dopo 10 tentativi il gioco ha termine
230-260	fine del gioco e scoperta della posizione dove è situato l'oro

Suggerimenti

- cambiare la forma della scacchiera (triangolare?)
- aggiungere delle ambiguità nell'indicare l'ubicazione dell'oro
- usare suoni diversi per segnalare la distanza dall'oro, note acute quando è vicino, gravi quando è lontano
- aggiungere suoni alla routine di vincita

```
1 REM FEBBRE DELL'ORO
7 POKE36879,15
8 PRINT""
10 A=INT(RND(1)*10)+1
20 B=INT(RND(1)*10)+1
30 FORZ=1TO10
35 PRINT"":PRINT
40 PRINT"HAI SOLO"
45 PRINT"      "10-Z;"TENTATIVI!"
50 PRINT"   DOVE E' L'ORO";
60 INPUTM:C=INT(M/10):D=M-C*10
70 IFA=CANDB=DTHEN230
130 PRINT:PRINT"  NON E' A";C;D
135 PRINT
140 PRINT"  TI DO' UN AIUTO"
150 PRINT"   VAI A ";
160 IFA>CTHENPRINTCHR$(28)"SUD";
170 IFA<CTHENPRINT"NORD";
180 IFB<DTHENPRINT"OVEST"
190 IFB>DTHENPRINT"EST"
200 NEXTZ
210 PRINT:PRINTCHR$(28)" IL TEMPO E' P
ASSATO"
215 PRINT"    L'ORO ERA A";A;B
220 END
230 PRINT"MOLTO BENE !!
!"
240 S=RND(1)*10000
245 PRINT""
250 PRINT"HAI TROVATO \";INT(S/Z);
255 FORT=1TO555:NEXTT
260 GOTO250
```

Lista simboli grafici

```
  8 : 1  SHIFT HOME =CHR$(147)

 40 : 6  CRSR← =CHR$(29)

 50 : 1  CRSR↑ =CHR$(17)

215 : 1  CRSR↑ =CHR$(17)

230 : 2  CRSR↑ =CHR$(17)
      5  CRSR← =CHR$(29)
```

ASTEROIDI

In questo programma di Burbidge voi siete alla guida di una nave stellare. Improvvisamente vi trovate in mezzo ad uno sciame di asteroidi. Soltanto voi potete salvare la nave dalla distruzione. Voi siete il quadretto nell'angolo in alto a sinistra dello schermo. Appena dato il RUN vedrete gli asteroidi comparire in diverse posizioni. Usando i tasti di controllo (4 = a sinistra, 6 = a destra) vi dovete tenere lontano dagli asteroidi (i quali sono costituiti da masse gelatinose).

Mano a mano che il gioco procede la difficoltà aumenta, aumenta il numero degli asteroidi ed il loro colore varia.

```
1 REM ASTEROIDI DI ADAM BURBIDGE
2 REM RIVISTO DA COMINI RENATO
3 REM MEMORIA 3,5 K
4 SC=0:HS=0
5 PRINT"♥"
10 FORJ=1TO20:PRINT:NEXT
11 POKE36879,255
40 POKE7746,102
43 B=38466
45 POKE38466,4
50 FORT=1TO300:NEXT
60 A=7746
65 POKEA,32
70 GETA$:IFA$="4"THENPOKEA-21,32:POKEA
-22,32:POKEA-32,32:A=A-1:B=B-1
71 IFA$="6"THENPOKEA-32,32:POKEA-22,32
:POKEA-21,32:A=A+1:B=B+1
72 POKEA-21,32
73 POKEA-22,32
74 POKEA-32,32
75 POKEA,102
76 POKEB,4
77 FORT=1TO100:NEXT
```

```
78 IF PEEK(A+22)=81THEN200
79 SC=SC+1
80 X=INT(RND(1)*506)+7680
89 POKEX+30720,6
90 POKEX,81
93 IFSC>400THEN500
95 IFSC>300THENGOSUB400
96 IFSC>200THENGOSUB350
100 PRINT
110 GOTO65
200 POKE36877,220
210 FORL=15TO0STEP-1
220 POKE36878,L
230 FORM=1TO50
240 NEXTM
250 NEXTL
260 POKE36877,0
270 POKE36878,0
280 PRINT" ■■ASTRONAVE DISTRUTTA!■■"
285 PRINT"      PUNTI=";SC
286 IFSC>HSTHENPRINTCHR$(28)"■■■■MAX.
PUNTEGGIO=";SC:HS=SC
287 IFSC<HSTHENPRINT"MAX.PUNTEGGIO=";H
S
290 FORT=1TO5000:NEXT
295 SC=0
300 GOTO5
350 Y=INT(RND(1)*506)+7680
360 POKEY,81
370 POKEY+30720,2
380 RETURN
400 D=INT(RND(1)*506)+7680
410 POKED,81
420 POKED+30720,5
430 RETURN
500 PRINT"        ■■MOLTO BENE■■"
510 PRINTCHR$(28)"  ■MISSIONE COMPLETA
TA■"
520 POKE36878,13
530 POKE36876,135
540 FOR O=241TO180STEP-1
550 FORM=1TO5
555 POKE36876,O
```

```
560 NEXT M
570 NEXT O
580 POKE36876,0
590 POKE36878,0
600 END
```

Lista simboli grafici

```
   5 : 1   SHIFT HOME =CHR$(147)

 280 : 1   REVERSE =CHR$(18)
       1   SHIFT REVERSE =CHR$(146)

 286 : 2   CRSR↑ =CHR$(17)
       3   CRSR← =CHR$(29)

 500 : 1   REVERSE =CHR$(18)
       1   SHIFT REVERSE =CHR$(146)

 510 : 1   REVERSE =CHR$(18)
       1   SHIFT REVERSE =CHR$(146)
```

MUSIC — VIC

Questo non è un gioco ma un piacevole passatempo che ci potrà accompagnare durante le ore "programmative". Caricato il programma e dato il Run si udirà, come per incanto, la ormai famosa musica di STAR WARS. La melodia continuerà senza interruzioni e noi potremo, nel frattempo, comporre altri programmi e lanciarli senza che questa subisca la benchè minima interruzione. Inutile stare qui a spiegare il funzionamento di questa routine in linguaggio macchina, basti dire che essa fa uso di un interrupt chiamato ogni sessantesimo di secondo. Per interrompere la melodia basta dare il RUN STOP + RESTORE e per riascoltarla basta SYS 830.

Attenzione perchè essendo la routine immagazzinata nel buffer della cassetta l'uso del tape ne compromette il funzionamento, quindi non bisogna caricare programmi da cassetta.

```
 10 -------VIC-MUSIC--------
 60 POKE36878,15:
 70 FORI=830TO976:READC:POKEI,C:NEXT
 80 SYS830
 100 DATA120,169,5,141,60,3,169,6,141,6
1,3,169,133,133,0,169,3,133,1,169,93,14
1,20,3,169,3
 110 DATA141,21,3,88,96,206,61,3,208,28
,72,152,72,172,60
 150 DATA3,200,177,0,141,61,3,200,177,0
,201
 160 DATA1,240,12,141,12,144,140,60,3,1
04
 170 DATA168,104,76,191,234,160,255,208
,243
 190 DATA15,201,15,201,15,201,60,215,60
,228,15
 200 DATA225,15,223,15,219,60,235,35,22
8,15
```

```
 210 DATA225,15,223,15,219,60,235,35,22
8,15
 220 DATA225,15,223,15,225,60,219,35,20
1,15
 230 DATA201,60,215,60,228,15,225,15,22
3,15
 240 DATA219,60,235,35,228,15,225,15,22
3,15
 250 DATA219,60,235,35,228,15,225,15,22
3,15
 260 DATA219,60,235,1,1
```

GUARDIANO DEL TEMPO

In questo gioco-sfida, scritto da ADAM Burbidge di Crawley, tu devi recuperare le 6 parti della chiave del tempo. La tua astronave parte con 700 punti di energia ed ogni mossa ti costa 5 punti.

Appena dato il RUN appare il messaggio "sto creando la galassia", il VIC sta cioè creando l'universo dove giocherai. Dopo un breve ritardo, lo schermo sarà cancellato e sarà il momento delle decisioni.

Il VIC vi chiederà spesso le coordinate, stampando:

QUADRANTE (inserisci un numero tra 1 e 6)
SETTORE (inserisci un numero tra 1 e 6)
SETTORE (inserisci un numero tra 1 e 6)
TEMPO (inserisci zero o uno)

Quando atterri su un pianeta ostile il VIC stamperà "nave all'attacco" e "energia" più il valore di quest'ultima. Devi allora necessariamente e nel minor tempo possibile premere un tasto per fuggire dal pianeta ostile, infatti ogni combattimento consuma energia.

Di tanto in tanto troverai delle parti della chiave di Chronos, è lo scopo del gioco. Per vincere devi trovare tutte le sei parti della chiave e devi trovarle prima che l'energia della nave sia finita.

```
 2 REM GUARDIANO DEL TEMPO
 3 REM DI ADAM BURBIDGE
 4 REM RIVISTO DA COMINI RENATO
 5 DIMK%(4),Z$(1),A%(6,6,6,1)
 6 K%(3)=700:F=75
 10 PRINT""
 20 PRINT"CREO LA GALASSIA"
 30 B=INT(RND(1)*6)
 40 C=INT(RND(1)*6)
 50 D=INT(RND(1)*6)
 60 E=INT(RND(1)*2)
 70 A%(B,C,D,E)=2
 75 F=F-1:IFF<0THEN85
 80 GOTO30
 85 PRINT""
 90 GN=INT(RND(1)*2)+1
 100 PRINT"PARTI A GLODNOB"
 105 PRINT
 110 PRINT"QUADRANTE 5"
 115 PRINT
 120 PRINT"SETTORE 3,4"
 125 PRINT
 130 PRINTCHR$(28)"TEMPO"GN*100
0:PRINT"ANNI TERRESTRI"
 135 PRINT
 140 PRINT"    COORDINATE"
 150 INPUT"QUADRANTE";G
 160 INPUT"SETTORE";H
 165 PRINTCHR$(28)
 170 INPUT"SETTORE";I
 180 INPUT"TEMPO (0/1)";J
 185 K%(3)=K%(3)-5
 190 IFA%(G,H,I,J)=1THENPRINT"SEI SU
GLODNOB":GOTO90
 200 IFA%(G,H,I,J)<>1THEN300
 210 PRINT"PROFONDO SPAZIO":GOTO14
0
 300 PRINT" SEI SU UN PIANETA"
 310 PRINT"QUADRANTE"G
 320 PRINT"SETTORE"H;;I
 330 PRINT"TEMPO"J*INT(RND(1)
*7)
 340 K%(1)=INT(RND(1)*20)+1
```

```
 350 IFK%(1)>5THEN370
 360 PRINT"DISABITATO":GOTO
140
 370 IFK%(1)>10THEN400
 380 PRINT"ABITANTI OSTILI":GOT
O500
 390 PRINT"HAI"
 400 PRINT" PARTE DELLA CHIAVE DI
CHRONOS"
 410 K%(2)=K%(2)+1
 420 PRINT" HAI ORA"K%(2)"PARTI DELLA
CHIAVE"
 430 IFK%(2)=6THENPRINT"HAI VINT
O":END
 440 GOTO140
```

```
500 GETZ$(1)
510 IFZ$(1)=""THEN540
520 GOTO140
540 PRINT"    ■▩NAVE ALL'ATTACCO!■"
550 K%(3)=K%(3)-INT(RND(1)*50)+1
560 PRINT"▩  ENERGIA NAVE:▩"CHR$(28)K%
(3)"■"
565 IFK%(3)<1THENPRINT"  NAVE DISTRUTT
A":END
570 FORT=1TO200:NEXT
580 GOTO500
```

Lista simboli grafici

```
 10 : 1  SHIFT HOME =CHR$(147)

 20 : 1  REVERSE =CHR$(18)
      3  CRSR← =CHR$(29)
      1  SHIFT REVERSE =CHR$(146)

 85 : 1  SHIFT HOME =CHR$(147)

100 : 1  REVERSE =CHR$(18)
      4  CRSR← =CHR$(29)
      1  SHIFT REVERSE =CHR$(146)

110 : 7  CRSR← =CHR$(29)

120 : 9  CRSR← =CHR$(29)

130 : 7  CRSR← =CHR$(29)
      2  REVERSE =CHR$(18)
      1  CRSR↑ =CHR$(17)
      5  CRSR← =CHR$(29)

150 : 5  CRSR← =CHR$(29)

160 : 7  CRSR← =CHR$(29)
```

```
170 : 7   CRSR← =CHR$(29)

180 : 3   CRSR← =CHR$(29)

190 : 1   REVERSE =CHR$(18)

210 : 1   CRSR↑ =CHR$(17)
      3   CRSR← =CHR$(29)

310 : 1   REVERSE =CHR$(18)
      5   CRSR← =CHR$(29)

320 : 7   CRSR← =CHR$(29)

330 : 9   CRSR← =CHR$(29)

360 : 1   REVERSE =CHR$(18)
      5   CRSR← =CHR$(29)
380 : 1   REVERSE =CHR$(18)
      4   CRSR← =CHR$(29)
      1   CRSR↑ =CHR$(17)

390 : 1   REVERSE =CHR$(18)
      9   CRSR← =CHR$(29)

400 : 1   REVERSE =CHR$(18)
      1   CRSR↑ =CHR$(17)
      8   CRSR← =CHR$(29)
      1   SHIFT CRSR← =CHR$(157)
      1   SHIFT REVERSE =CHR$(146)

420 : 1   CRSR↑ =CHR$(17)
      6   CRSR← =CHR$(29)

430 : 1   REVERSE =CHR$(18)
      5   CRSR← =CHR$(29)

540 : 1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)

560 : 1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
```

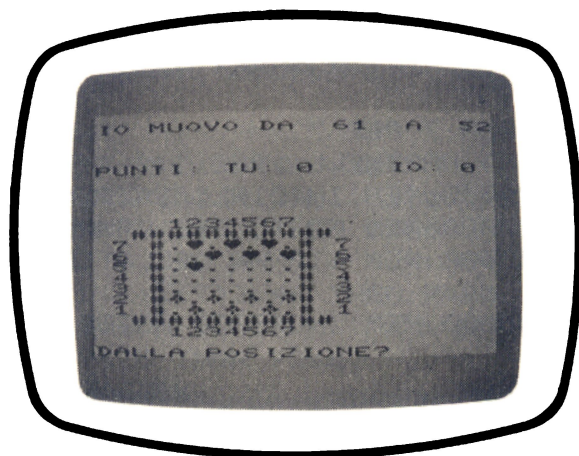
DAMA 7 PER 7 (I VECCHI DEL 49)

Il gioco è come la dama tranne che, come ho detto, si gioca su una scacchiera 7 per 7 (da qui il nome del gioco).

I pezzi si muovono come a dama, cioè diagonalmente di una casella, saltando sopra un avversario in una casella vuota al di là della stessa per mangiarlo. Le principali differenze con la dama sono che, a parte la scacchiera, i pezzi possono muoversi a piacere avanti ed indietro, non vi sono damoni (ma è come se ogni pezzo lo fosse) ne' salti multipli. Il computer muove i Cuori sullo schermo e tu i Picche. Tu muovi mettendo il n° della casella da cui ti stai muovendo (scrivendo prima ordinata e poi ascissa ed infine dando RETURN) poi il n° della casella in cui vai.

Il computer registra il punteggio, ti dice (prima di muovere i suoi pezzi) in quale casella sta andando e pone fine al gioco appena un giocatore riesce a catturare 5 pezzi avversari.

Il programma presenta alcune tecniche strutturate.

La linea 30 manda il controllo ad una subroutine situata in fondo al programma che inizializza le variabili, poi il computer entra nella sua routine di gioco (linea 7000).

Fatta la mossa torna alla linea 40 e va alla subroutine della linea 8000 che ti mostra la scacchiera, da qui va alla subroutine 6000 per accettare la mossa del giocatore, torna di nuovo alla 8000 a ripristinare la scacchiera e poi torna indietro per eseguire la successiva mossa del computer e così via.

Il vantaggio di lavorare in questo modo con una subroutine specifica per ogni sezione del programma, è che tu puoi facilmente cambiare le parti che non ti sembrano soddisfacenti senza dover affrontare il rischio di esaurire le linee o di "perdere la strada". Ponendo l'assegnazione delle variabili alla fine hai il vantaggio di poterne aggiungere di nuove quando ne appare la necessità, senza alcun problema. La disposizione del programma è semplice.

Remarks

1-1000 anello principale che passa il controllo alle varie subroutine

6000-6990 accetta le mosse del giocatore ed aggiorna il suo punteggio

7000-7470 mossa del computer e aggiornamento del suo punteggio

8000-8990 stampa della scacchiera e del punteggio e verifica di fine gioco

9000-9140 inizializzazione

Sebbene questo metodo di programmazione non sia il più idoneo per risparmiare memoria esso rende il compito della ricerca degli errori e dello sviluppo del programma molto più semplice.

Se tu volessi sapere di più sulla struttura e lo svolgimento di un programma faresti bene a leggere il "Basic con stile" di Paul A. Nagin e Henry F. Ledgard pubblicato dalla Hayden Book Company inc.

```
 10 REM DAMA 7 PER 7
 20 REM (C) HARTNELL 1982
 30 GOSUB9000:REM INIZIALIZZAZIONE
 40 GOSUB7000:REM MOSSE DEL COMPUTER
 60 GOSUB8000:REM STAMPA DELLA SCACCHIE
RA
 70 GOSUB6000:REM INGRESSO MOSSE DEL GI
OCATORE
 1000 GOTO40
 6000 PRINT
 6050 PRINT"DALLA POSIZIONE";
```

```
6060 INPUT M
6070 PRINT" ALLA POSIZIONE";
6080 INPUT N
6090 H(N)=120
6095 IF ABS(M-N)=22 OR ABS(M-N)=18 THE
N H((M+N)/2)=46
6097 IF ABS(M-N)=22 OR ABS(M-N)=18 THE
N ME=ME+1
6100 H(M)=46
6990 RETURN
7000 REM MOSSE DEL COMPUTER
7010 FORA=76TO12STEP-1
7020 IF H(A)<>211THEN7060
7030 FOR B=1 TO 4
7032 IF A<28 AND B<3 THEN7050
7033 IF A>60 AND B>2 THEN7060
7035 Q=2*Z(B)
7040 IF H(A+Z(B))=120 AND H(A+Q)=46 TH
EN7070
7050 NEXT B
7060 NEXT A
7065 GOTO 7300
7070 H(A+Z(B))=46
7080 H(A)=46
7090 H(A+Q)=211
7092 Y=A+Q
7093 X=A
7095 IT=IT+1
7100 RETURN
7200 NEXT A
7300 REM MOSSA CASUALE
7310 Y=0
7320 Y=Y+1
7330 K=INT(RND(1)*66)+12
7340 IF H(K)<>211 AND Y<100 THEN 7320
7350 IF H(K)<>211 THEN 7460
7360 FOR T=1 TO 4
7370 IF H(K+Z(T))=46 THEN 7400
7380 NEXT T
7390 IF Y<70 THEN 7310
7395 GOTO 7460
7400 H(K+Z(T))=211
7410 H(K)=46
```

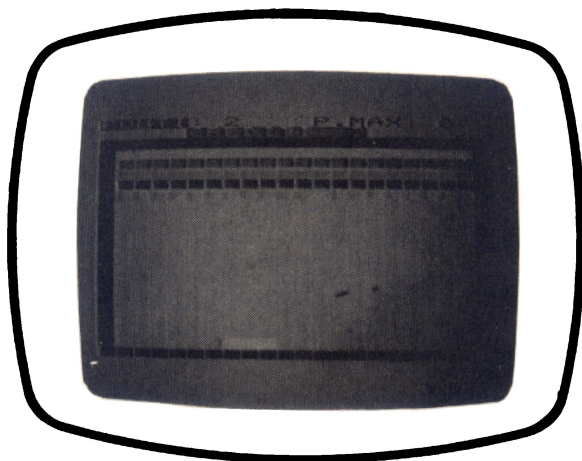
```
 7415 X=K:Y=K+Z(T)
 7420 RETURN
 7460 PRINT"IO LASCIO"
 7470 END
 7990 RETURN
 8000 REM PULITURA SCHERMO
 8010 PRINT"♥"
 8015 PRINT"IO MUOVO DA ";X;" A ";Y
 8017 PRINT:PRINT
 8020 PRINT "PUNTI: TU:";ME;"   IO:";IT
 8040 PRINT:PRINT:PRINT
 8050 PRINT"████1234567"
 8055 PRINT"██#########"
 8060 FOR J=70 TO 10 STEP -10
 8070 A=H(J+1):B=H(J+2):C=H(J+3):D=H(J+
4):E=H(J+5):F=H(J+6):G=H(J+7)
 8080 PRINT ;J/10;"#";CHR$(A);CHR$(B);C
HR$(C);CHR$(D);CHR$(E);CHR$(F);CHR$(G);
"#";J/10
 8090 NEXT J
 8100 PRINT"██#########"
 8110 PRINT"████1234567"
 8120 IF IT=5 OR ME=5 THEN 8140
 8130 RETURN
 8140 IF IT=5 THEN PRINT"HO VINTO IO!"
 8150 IF ME=5 THEN PRINT"HAI VINTO TU!"
 8999 END
 9000 REM INIZIALIZZAZIONE
 9010 IT=0:ME=0
 9020 DIM H(99),Z(4)
 9030 FOR A=1 TO 99
 9040 H(A)=0
 9050 IFA>77ORA=70ORA=60ORA=68ORA=69ORA
=50ORA=59ORA=58ORA=40ORA=49ORA=48 THEN
GOTO 9090
 9055 IFA=30ORA=38ORA=39ORA=20ORA=28ORA
=29ORA<11 THEN GOTO 9090
 9060 H(A)=46:REM .
 9070 IF A=72 OR A=74 OR A=76 OR A=61 O
R A=63 OR A=65 OR A=67 THEN H(A)=211
 9080 IF A=21 OR A=23 OR A=25 OR A=27 O
R A=12 OR A=14 OR A=16 THEN H(A)=120:RE
M X
```

```
9090 NEXT A
9100 FOR A=1 TO 4:READ B:Z(A)=B:NEXT A
9110 DATA -11,-9,11,9
9200 RETURN
```

Lista simboli grafici

```
8010 : 1  SHIFT HOME =CHR$(147)

8050 : 4  CRSR← =CHR$(29)

8055 : 2  CRSR← =CHR$(29)

8100 : 2  CRSR← =CHR$(29)

8110 : 4  CRSR← =CHR$(29)
```

BREACKOUT

Il gioco propone una versione per VIC 20, nella configurazione minima di memoria dell'ormai famoso breackout: guidare una pallina che rimbalza verso la distruzione completa di un muro. Il movimento della paletta è stato codificato in linguaggio macchina per ovvie esigenze di velocità.

Remarks

10-60	il codice macchina viene depositato mediante un normale ciclo che include READ DATA POKE
70	nella locazione 674 è memorizzata l'ultima posizione della paletta mentre nelle locazioni libere in pagina zero (0 e 1) vi è l'indirizzo dove prelevare il codice del tasto premuto
80-90	da S1 a S4 sono tutti i registri per il controllo dello schermo; T3 è l'indirizzo del generatore di tonalità più bassa; VO è l'indirizzo del registro volume. Si settano i colori per il bordo e lo sfondo
110-160	presentazione e istruzioni

170-175	é in pratica il livello di difficoltà: più è alto il numero di palette a disposizione più è facile abbattere completamente il muro. Una risposta non corretta dà automaticamente una sola paletta
180	Y contiene la posizifone verticale della pallina, DY la sua direzione (−1 equivale a IN ALTO e 1 a IN BASSO)
190	azzeramento del punteggio
200-230	crea il bordo del campo
240-260	stampa delle 5 file di mattoncini ciascuna delle quali ha un colore ed un punteggio diverso. D è il numero di palette in credito, oltre a quella con cui si sta giocando
270	situazione punteggio, punteggio massimo e palette disponibili
280	é la routine in linguaggio macchina che preleva la eventuale mossa e sposta di conseguenza la paletta. La velocità di quest'ultima dipende dal numero di istruzioni SYS 680 che si trovano all'interno del programma compreso tra la linea 260 e la linea 360
290	cancella la vecchia posizione della pallina
300-340	viene calcolata la sua nuova posizione controllando se vi è un rimbalzo laterale (DX = −DX), se la pallina ha urtato il fondo o qualche mattone
355	con SYS 736 parte la seconda routine ASSEMBLER che conta quanti mattoni restano ancora da demolire; se non ve ne sono più, viene messo un 1 nella locazione 673, che altrimenti contiene 0
360	altra mossa del giocatore, quindi si ritorna alla stampa della pallina, al punteggio e così via
400-420	la pallina ha urtato il fondo. Generazione del suono di commento. Viene decrementato il numero di palline che restano: se queste sono un numero maggiore o uguale a zero, potete ancora continuare (invertendo i valori di direzione X e DX, Y e DY), altrimenti si salta alla linea 700
450-500	sono stati distrutti tutti i mattoni e ciò viene sottolineato visivamente e musicalmente. Viene anche generato un bonus che tiene conto del punteggio e del livello di difficoltà

510-720 scritte di conclusione e la richiesta di una nuova partita

3000-3020 questa subroutine genera un valore casuale per X e DX, se DX = −1 la pallina parte a sinistra, altrimenti (DX = 1) parte a destra

```
 10 FORI=1TO55:READA:POKE679+I,A:NEXT
 20 FORI=1TO20:READA:POKE735+I,A:NEXT
 30 DATA172,162,2,169,32,153,205,31,153
,206,31,153,207,31,162,0,161,0,205,163,
2,208,7
 40 DATA192,2,240,3,136,161,0,205,164,2
,208,5,192,19,240,1,200,169,224,153,205
,31,153
 50 DATA206,31,153,207,31,140,162,2,96
 60 DATA162,0,189,0,30,201,204,240,10,2
32,224,255,208,244,169,1,141,161,2,96
 70 POKE0,197:POKE1,0:POKE673,0:POKE674
,10:POKE675,33:POKE676,34
 80 A$="██████████████":S1=36864:S2=1+S1
:S3=S2+1:S4=S3+1:T3=36876:VO=T3+2:T2=T3
-1
 90 PRINT"██":POKEVO+1,253
 110 PRINT"█            ██BRECKOUT██"
 120 PRINT"██  DISTRUGGI TUTTI  I  MATT
ONI DEL MURO, OGNI FILA DA' UN PUNTEGGI
O
 130 PRINT"██ CON ███ VAI A SINISTRA"
 140 PRINT"█ CON ███ VAI A DESTRA"
 150 PRINT"██    █"CHR$(28)" PREMI UN TA
STO █"
 160 GETB$:IFB$=""THEN160
 170 PRINT"████"CHR$(28)" QUANTE PALETT
E VUOI█":INPUT"    (1,2,3,4)";L
 175 IFL<1ORL>4THENL=1
 180 POKEVO+1,239:POKEVO,15:Y=20:DY=-1:
GOSUB3000:PRINT"██":IFPU>PMTHENPM=PU
 190 PU=0
 200 FORI=7680+44TO7680+44+21:POKEI,224
 205 POKEI+30720,2:NEXT
```

```
 210 FORI=3TO22:POKE7680+22*I,208:POKE3
8400+22*I,2:POKE7701+22*I,207
 220 POKE38421+I*22,2:NEXT
 230 FORI=0TO21:POKE8164+I,207:POKE3888
4+I,2:NEXT
 240 FORI=3TO7:FORJ=1TO20
 250 POKE7680+22*I+J,204:POKE38400+22*I
+J,I:NEXTJ,I:D=L-1
 260 POKE7680+X+22*Y,81
 270 PRINT"■■PUNTI■:"PU;TAB(12)"P.MAX:
"PM"■■■■■■■CREDITI:"D"■"
 280 SYS680
 290 POKE7680+X+22*Y,32
 300 X=X+DX
 310 IFX<2ORX>19THENDX=-DX:POKET3,220
 320 Y=Y+DY:IFY=22THEN400
 330 IFPEEK(7658+X+22*Y)=204THENDY=1:PO
KET3,230:POKE7658+X+22*Y,32:PU=PU+(10-Y
)
 340 IFPEEK(7702+X+22*Y)=224ORPEEK(7658
+X+22*Y)=224THENDY=-DY:POKET3,240
 350 POKET3,0
 355 IFPU>190THENSYS736:IFPEEK(673)=1TH
EN450
 360 SYS680:GOTO260
 400 POKEVO,15:FORI=230TO160STEP-.15:PO
KET2-1,I:NEXT:POKET2-1,0
 410 D=D-1:IFD>-1THENY=20:DY=-1:GOSUB30
00:GOTO260
 420 GOTO700
 450 PRINT"■":POKEVO+1,27:FORI=160TO240
STEP.5:POKET3,I:B%=RND(1)*505+1
 460 POKE7680+B%,160:POKE38400+B%,INT(R
ND(1)*7)+1
 470 NEXT:POKET3,0:O%=(PU/(L*3)+.5):I=0
:J=0
 480 J=J+1:FORI=0TO22:POKES1,12+I:POKES
2,38+I:POKES3,150-I:POKES4,174-I*2:POKE
T3,220+I:NEXT
 490 FORI=22TO0STEP-1:POKES1,12+I:POKES
2,38+I:POKES3,150-I:POKES4,174-I*2:POKE
T3,220+I:NEXT
 500 IFJ<5THEN480
```

```
 510 POKET3,0:PRINT"VINCI UN BONUS
 DI "Q%" PUNTI IL PUNTEGGIO E' ORA "PU+
Q%:PU=PU+Q%
 520 FORI=0TO4000:NEXT
 530 PRINT"COMPLIMENTI, CE L'HAI F
ATTA!!! IL TUO PUNTEGGIO E' STATO DI
"PU"."
 540 IFPU>PMTHENPRINT"BRAVO PERCHE
' E' ANCHE IL MIGLIORE !"
 550 GOTO710
 700 PRINT"N. PALETTE "D+1:POKET
3,0:POKET2,0
 710 POKE198,0:PRINTA$"    ANCORA (S/N)
";
 720 INPUTR$:IFLEFT$(R$,1)="S"THEN180
 740 PRINT""CHR$(28)"C
IAO":END
 3000 X=INT(RND(1)*20)+1:IFX<2ORX>19THE
N3000
 3010 DX=INT(RND(1)*3)-1:IFDX=0THEN3010
 3020 RETURN
```

Lista simboli grafici

```
  80 : 1   HOME =CHR$(19)
       11   CRSR↑ =CHR$(17)
       1   CRSR← =CHR$(29)

  90 : 1   SHIFT HOME =CHR$(147)

 110 : 1   CRSR↑ =CHR$(17)
       1   REVERSE =CHR$(18)
       1   SHIFT REVERSE =CHR$(146)

 120 : 2   CRSR↑ =CHR$(17)

 130 : 2   CRSR↑ =CHR$(17)

 140 : 1   CRSR↑ =CHR$(17)
```

```
150 : 2   CRSR↑ =CHR$(17)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)

170 : 1   SHIFT HOME =CHR$(147)
      4   CRSR↑ =CHR$(17)

180 : 1   SHIFT HOME =CHR$(147)

270 : 1   HOME =CHR$(19)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
      1   HOME =CHR$(19)
      1   CRSR↑ =CHR$(17)
      5   CRSR← =CHR$(29)

450 : 1   SHIFT HOME =CHR$(147)

510 : 4   CRSR↑ =CHR$(17)

530 : 1   SHIFT HOME =CHR$(147)
      1   CRSR↑ =CHR$(17)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)

540 : 1   CRSR↑ =CHR$(17)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)

700 : 1   HOME =CHR$(19)
      1   CRSR↑ =CHR$(17)
      5   CRSR← =CHR$(29)

740 : 1   SHIFT HOME =CHR$(147)
      6   CRSR↑ =CHR$(17)
      9   CRSR← =CHR$(29)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
```

IL GIOCO DELLA SCIMMIA

È questo un gioco molto giovane e noto. Si tratta di "scimmiottare" il calcolatore esercitando la nostra memoria visiva e sonora.

Vediamo quale è il principio di funzionamento: si accendono uno per volta in 4 diverse posizioni dello schermo, altrettanti quadrati, ciascuno di un colore diverso. Sono stati usati il verde il porpora il blu ed il giallo, cui corrispondono i tasti 5, 6, 7, 8 (senza schiacciare il tasto ctrl); il colore dei singoli flashes è generato casualmente. In pratica bisogna solo ripetere la sequenza premendo i corrispondenti tasti coi colori: ci sono 4 tipi di gioco.

GIOCO 1 — create la sequenza

Dopo aver ripetuto la prima nota che il VIC vi suggerisce siete voi a dover aggiungere una nuova nota. Continuando in questo modo dovete ripetere le prime due note e quindi aggiungerne ancora una, e così via finchè non avrete esaurito il numero di passi stabiliti per ciascun livello di difficoltà (tabella 1).

GIOCO 2 — ripetete la sequenza

Il VIC genera la prima nota; ripetetela correttamente ed il VIC ne aggiungerà una seconda, sempre secondo una generazione casuale, e così via finchè non sarà stato raggiunto il numero di note corrispondente alla sequenza scelta.

GIOCO 3 — la sequenza che cambia

È molto simile al gioco 2. L'unica differenza è che il VIC non ripete la sequenza generata precedentemente, ma inizia sempre con una nuova cui di volta in volta viene aggiunta una nuova nota.

GIOCO 4 — singoli Flashes

Il VIC genera una sola nota che bisogna ripetere entro un secondo.

Remarks

40-200	inizializzazione. Scelta del tipo di gioco e difficoltà
220-280	istruzioni relative al primo gioco
300-350	quelle relative al secondo

370-390 quelle relative al terzo

410-430 quelle relative al quarto

450-520 subroutine che genera il suono ed il relativo quadrato colorato per ogni flash

540-610 subroutine per la ripetizione della sequenza

630-660 scritte finali nel caso di errore

670-700 musica di accompagnamento

720-760 commenti in caso di vittoria, punteggio parziale e totale, le prime note della QUINTA SINFONIA

780-860 possibilità di rivedere l'ultima sequenza

880-940 proposta per una nuova partita

960-980 svuotamento dello STACK.

Principali variabili usate

BK Durata delle note
BL. 11 discese del cursore più il codice del porpora
CL. I 4 colori corrispondenti alle posizioni di ciascun quadrato
D Indirizzo del generatore di tonalità alta
D1 Intervallo di tempo tra due flash
D2 Tempo, in secondi, che è a disposizifone per la risposta
D3 Memorizza il tempo di risposta per confrontarlo poi con D2
G Numero di partite con lo stesso tipo di gioco
K (N) Posizioni dei quadrati colorati nell'N-esimo flash della sequenza
N1 Numero massimo di flash per quella sequenza
P Numeri di flash nel gioco 4
Q Flag prima dell'uscita da una subroutine
VO Indirizzo registro volume
X Numero del gioco

Livelli di difficoltà

tasto funzione	N° note	intervallo tra 2 note	tempo per rispondere
F1	4	100	5 sec.
F3	8	50	3 sec.
F5	16	10	2 sec.
F7	32	1	1 sec.

```
 10 REM IL GIOCO DELLA SCIMMIA
 40 POKE36879,13:PRINT""
 50 PRINT"QUATTRO TIPI DI GIOCO:
1.UNA TUA SEQUENZA   2.RIPETENDO LA"
 60 PRINT"SEQUENZA":PRINT" 3.CAMBIA
NDO SEQUENZA  4.SINGOLI FLASH"
 70 PRINT"    CHE NUMERO ? ";
 80 GETA$:IFA$=""THEN80
 90 X=VAL(A$):PRINTX
 100 PRINT" PER INCREMENTARE IL  LIV
ELLO DI DIFFICOLTA'PREMI F1,F3,F5,F7
"
 110 GETA$:IFA$=""THEN110
 120 IFA$=CHR$(133)THEND1=100:N1=4:D2=5
 130 IFA$=CHR$(134)THEND1=50:N1=8:D2=3
 140 IFA$=CHR$(135)THEND1=10:N1=16:D2=2
 150 IFA$=CHR$(136)THEND1=1:N1=32:D2=1
 160 B$="":C$="PORP. VERDE
BLU GIALLO"
 170 DIMK(32):VO=36878:D=36876:S=7790:C
O=38510:G=0:DD=0
 180 NM=1:G=G+1:Q=0
 190 PRINT""B$C$:W=RND(-TI)
 200 ONXGOTO220,300,410
 220 PRINT"IL VIC SCEGLIE IL PRIMO COL
ORE":N=1:K(1)=INT(RND(1)*4):GOSUB450
 230 GOSUB540:IFQ<>0THEN960
 240 NM=NM+1:N=NM:PRINT"AGGIUNGI UN N
UOVO         COLORE":D3=TI
```

```
 250 GETA$:IFA$=""ANDTI-D3<D2*60THEN250
 260 IFTI-D3>D2*50THEN630
 270 K(N)=VAL(A$)-5:GOSUB450:GOSUB540:I
FQ<>0THEN960
 280 GOTO240
 300 N=NM:K(N)=INT(RND(1)*4):GOSUB450
 310 GOSUB540:IFQ<>0THEN960
 320 FORI=1TO800:NEXT
 330 FORN=1TONM:PRINT"█▌IL VIC TI DA' Q
UESTA SEQUENZA":GOSUB450:NEXTN
 340 NM=NM+1:GOTO300
 350 FORI=1TO800:NEXT
 370 FORN=1TONM:PRINT"█▌IL VIC TI DA' Q
UESTA SEQUENZA":K(N)=INT(RND(1)*4)
 380 GOSUB450:NEXTN:GOSUB540:IFQ<>0THEN
960
 390 FORI=1TO800:NEXT:NM=NM+1:GOTO370
 410 P=1
 420 D2=1:N=1:K(N)=INT(RND(1)*4):GOSUB4
50:GOSUB540:IFQ<>0THEN960
 430 FORI=1TO400:NEXT:NM=1:N=1:P=P+1:GO
TO420
 450 POKEVO,15:POKED,135+K(N)*25:PRINT"
█"B$+"█████████PASSO NUM.:";
 460 IFX=4THENPRINTP:GOTO480
 470 PRINTN
 480 FORJ=0TO4:FORL=CO+J*22+6*K(N)TOL+3
:POKEL,K(N)+4:NEXTL,J
 490 FORJ=0TO4:FORI=S+J*22+6*K(N)TOI+3:
POKEI,160:NEXTI,J
 500 FORJ=0TOD1:NEXT:POKED,0
 510 PRINT"█"B$C$
 520 RETURN
 540 FORN=1TONM:PRINT"█  RIPETI LA SEQU
ENZA":D3=TI
 550 GETA$:IFA$=""ANDTI-D3<D2*60THEN550
 560 IFTI-D3>D2*55THENQ=1:RETURN
 570 Y=VAL(A$):IFY-5<>K(N)THENQ=2:RETUR
N
 580 IFY-5=K(N)THENGOSUB450:NEXTN
 590 IFX=4THENN=P+1
 600 IFN=N1+1THENQ=3:RETURN
 610 RETURN
```

```
 630 PRINT"HAI PERSO! NON SEI STATO
ABBASTANZA VELOCE":GOTO650
 640 PRINT"-ERRORE- HAI PERSO!"
 650 FORN=1TO1:NEXT:IFX=4THENNM=P
 660 PRINT"  "NM-1"   PASSI ERANO
   CORRETTI":PRINT" N. DI GIOCHI:"G
 670 POKEVO,15:RESTORE
 680 READW:IFW=-1THENPOKET+1,0:GOTO780
 690 READBK:POKET+1,W:FORI=1TOBK:NEXTI:
POKET+1,0:FORI=1TO10:NEXTI:GOTO680
 700 DATA175,100,175,100,175,100,151,50
0,0,300,163,100,163,100,163,100,147,500
,-1
 720 PRINT"COMPLIMENTI HAI FINITO
 CON"N1" PUNTI"
 730 PRINT" N. DI GIOCHI:"G:DD=DD+N1:P
RINT"IN TOTALE HAI"DD;
 740 PRINT"PUNTI":POKEVO,15:RESTORE
 750 READW:IFW=-1THEN780
 760 READBK:POKED,W:FORI=1TOBK:NEXTI:PO
KED,0:FORI=1TO10:NEXTI:GOTO750
 780 IFX=3ORX=4THEN880
 800 PRINT"VUOI VEDERE L'ULTIMA
SEQUENZA(S/N)"
 810 GETA$:IFA$=""THEN810
 820 IFA$<>"S"ANDA$<>"N"THEN810
 830 IFA$="S"THENPRINT""B$C$
 840 IFA$="N"THEN880
 850 IFN=N1+1THENNM=NM+1
 860 FORN=1TONM-1:GOSUB450:NEXTN
 880 PRINT"  VUOI CONTINUARE"
 885 PRINT"   1.ALLO STESSO MODO2
.CAMBIANDO LE"
 890 PRINT"CONDIZIONI":PRINT"3
. FINIRE":PRINT"   PREMI IL NUMERO"
 895 PRINT"DESIDERATO"
 900 GET A$:IFA$=""THEN900
 910 Y=VAL(A$):IFY<>1ANDY<>2ANDY<>3THEN
900
 920 ONYGOTO180,930,940
 930 RUN
 940 END
 950 REM -PULIZIA DELLO STACK-
```

```
960 IFQ=1THENQ=0:GOTO630
970 IFQ=2THENQ=0:GOTO640
980 IFQ=3THENQ=0:GOTO720
```

Lista simboli grafici

```
 50 : 1  SHIFT HOME =CHR$(147)
      2  CRSR↑ =CHR$(17)
      1  REVERSE =CHR$(18)
      1  SHIFT REVERSE =CHR$(146)
      2  CRSR↑ =CHR$(17)

 60 : 3  CRSR← =CHR$(29)
      2  CRSR↑ =CHR$(17)

 70 : 2  CRSR↑ =CHR$(17)
      1  REVERSE =CHR$(18)
      1  SHIFT REVERSE =CHR$(146)

100 : 2  CRSR↑ =CHR$(17)
      1  REVERSE =CHR$(18)
      3  CRSR← =CHR$(29)

160 : 10  CRSR↑ =CHR$(17)

190 : 1  SHIFT HOME =CHR$(147)

220 : 1  CRSR↑ =CHR$(17)

240 : 1  CRSR↑ =CHR$(17)
      1  CRSR← =CHR$(29)

330 : 1  CRSR↑ =CHR$(17)
      1  CRSR← =CHR$(29)
```

```
370 : 1   CRSR↑ =CHR$(17)
      1   CRSR← =CHR$(29)

450 : 1   HOME =CHR$(19)
      5   CRSR↑ =CHR$(17)
      3   CRSR← =CHR$(29)

510 : 1   SHIFT HOME =CHR$(147)

540 : 1   CRSR↑ =CHR$(17)

630 : 1   SHIFT HOME =CHR$(147)
      2   CRSR↑ =CHR$(17)

640 : 1   SHIFT HOME =CHR$(147)
      2   CRSR↑ =CHR$(17)

660 : 2   CRSR↑ =CHR$(17)

720 : 1   SHIFT HOME =CHR$(147)
      2   CRSR↑ =CHR$(17)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)

730 : 2   CRSR↑ =CHR$(17)

800 : 2   CRSR↑ =CHR$(17)
      1   CRSR← =CHR$(29)
      1   REVERSE =CHR$(18)
      5   CRSR← =CHR$(29)
      1   SHIFT REVERSE =CHR$(146)

830 : 1   SHIFT HOME =CHR$(147)

880 : 1   SHIFT HOME =CHR$(147)
      3   CRSR↑ =CHR$(17)

885 : 1   REVERSE =CHR$(18)
      4   CRSR← =CHR$(29)
      1   CRSR↑ =CHR$(17)
```

```
890 : 1   REVERSE =CHR$(18)
      4   CRSR← =CHR$(29)
      1   REVERSE =CHR$(18)
      1   CRSR↑ =CHR$(17)
      2   CRSR← =CHR$(29)
      3   CRSR↑ =CHR$(17)
      1   SHIFT REVERSE =CHR$(146)

895 : 5   CRSR← =CHR$(29)
```

SCRIVERE E CONVERTIRE GIOCHI

Questa sezione contiene importanti informazioni e può aiutarti molto quando dovrai programmare una scacchiera ed altri giochi per il tuo VIC. Ecco una scacchiera numerata per renderla più adatta all'uso che il computer ne dovrà fare. Tu puoi individuare ogni quadretto battendo prima il nu-

mero a sinistra (ordinata), ad esempio 3, e poi quello in alto (ascissa), ad esempio 4. In questo caso la riga numerata 3 (lungo il lato sinistro) e quella numerata 4 (colonna) in alto, si incontrano nella casella 34. Se tu desideri muovere un pezzo batti prima il numero della casella dove ti trovi (es. 55) e

poi il numero di quella dove desideri andare ed il computer capirà esattamente quello che vuoi fare. Questo sistema di numerazione della scacchiera ha un altro grande vantaggio rispetto ad un sistema che semplicemente numera le caselle in ordine: muovendoti in qualsiasi direzione, non importa dove tu sia sulla scacchiera, la differenza fra le caselle é la stessa.

Ora mi spiego: se ti muovi di una casella verso destra ed in alto (come normalmente si fa nella dama) tu ti muoverai ad esempio dalla casella 24 alla 35 o dalla 53 alla 64 o dalla 71 alla 82.

Nota che non ha importanza quale sia la tua posizione sulla scacchiera: la differenza fra il punto di partenza e quello di arrivo è sempre 11 (es. 35 − 24 = 11). Se ti muovi diagonalmente verso sinistra, ad esempio dalla 26 alla 35 o dalla 66 alla 75 o dalla 22 alla 31, la differenza sarà sempre 9. Questa prevedibilità rende relativamente semplice creare scacchiere che un computer possa trattare.

Immaginiamo che il computer abbia una dama sulla casella 24, può essere facilmente programmato per controllare ogni casella della scacchiera e ogni volta che trova uno dei suoi pezzi controlla se c'è un pezzo umano nella casella numerata (nel nostro caso la 24) più undici (cioè 24 + 11 = 35) e che la casella undici oltre la 35 (cioè la 46) sia vuota, il computer può allora saltare sopra la 35 mangiando il pezzo che vi si trova.

Questo è in sostanza il funzionamento di molti giochi di dama per computer (ma non quello di questo libro). Comunque una scacchiera simile a questa (solo che è 7 per 7 anzichè 10 per 10) è utilizzata dall'Othello.

Se volessi giocare a scacchi su una scacchiera di questo tipo potresti facilmente specificare i movimenti, ad esempio, di un cavallo sapendo che deve sempre saltare in caselle situate alla seguente distanza da quella di partenza: 21, 12, 8, 19, 21, 12, 19 o 8.

Verificalo tu stesso, se poni una moneta sulla 55, e la muovi come un cavallo, lavorando sulle relazioni matematiche fra le caselle di partenza e quelle d'arrivo troveresti che le differenze sono quelle sopracitate.

APPENDICE

Messaggi d'errore

BAD DATA... si riceve un dato di tipo stringa al posto di uno numerico

BAD SUBSCRIPT... ci si è riferiti ad un elemento di un array fuori dal rango richiesto con la DIM

CAN'T CONTINUE... il comando CONT non può essere eseguito, il programma non era partito oppure c'è stato un errore o si è appena corretta o inserita una linea

DEVICE NOT PRESENT... si è richiesto un I/O con OPEN, CLOSE, CMD, PRINT#, INPUT#, GET#, e la periferica non è disponibile

DIVISION BY ZERO... la divisione per zero è impossibile

EXTRA IGNORED... si risponde ad un INPUT con troppi dati, soltanto alcuni sono accettati

FORMULA TOO COMPLEX... un'espressione era troppo complessa, conviene spezzarla in due parti più semplici

ILLEGAL DIRECT... un INPUT può essere usato solo in un programma e non in modo diretto

ILLEGAL QUANTITY... un numero è usato come argomento di una funzione o di una altra operazione ed è fuori dai limiti consentiti

LOAD... ci sono dei problemi nel caricare da cassetta

NEXT WITHOUT FOR... dei cicli sono stati nidificati male, si ha un NEXT che non è preceduto dal relativo FOR

OUT OF DATA... si è usato un READ per leggere più dati di quanti ve ne siano nel blocco dati

OUT OF MEMORY... non c'è più RAM disponibile, questo può anche accadere se ci sono troppi FOR nidificati o troppi GOSUB concatenati

OVERFLOW... il risultato di un calcolo supera 1.70141884E+38

REDIM'D ARRAY... si cerca di dimensionare una variabile che è già stata dimensionata

REDO FROM START... si risponde ad un INPUT con dati di tipo errato, inserendo i dati esatti il programma continuerà

RETUR WITHOUT GOSUB... si incontra un RETURN senza che ci sia stata una chiamata di subroutine

STRING TOO LONG... una stringa non può contenere più di 255 caratteri

SYNTAX... in una linea c'è un errore di sintassi: parentesi mancanti, parole chiave inesatte ecc.

TYPE MISMATCH... errore nel tipo di dati: un numero è usato al posto di una stringa o viceversa

UNDEF'D STATEMENT...GOTO, GOSUB, THEN, RUN che si riferiscono a linee inesistenti

UNDEF'D FUNCTION... ci si riferisce ad una funzione utente che non è stata definita (DEF FN)

Mappa memoria relativa allo schermo

DOWN MODE Code Table

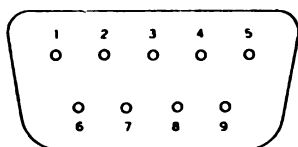
D7					0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
D6					0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
D5					0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
D4					0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
D3	D2	D1	D0		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0		POS	SP	ø	@	p	[graphic]	P			SP	[graphic]	[graphic]	P	SP	[graphic]
0	0	0	1	1		CURSOR DOWN	'	1	a	q	A	Q		CURSOR UP	[graphic]	[graphic]	A	Q	[graphic]	[graphic]
0	0	1	0	2		SVS ON	"	2	b	r	B	R		REV OFF	[graphic]	[graphic]	B	R	[graphic]	[graphic]
0	0	1	1	3			#	3	c	s	C	S			[graphic]	[graphic]	C	S	[graphic]	[graphic]
0	1	0	0	4			$	4	d	t	D	T			[graphic]	[graphic]	D	T	[graphic]	[graphic]
0	1	0	1	5			%	5	e	u	E	U			[graphic]	[graphic]	E	U	[graphic]	[graphic]
0	1	1	0	6			&	6	f	v	F	V			[graphic]	[graphic]	F	V	[graphic]	[graphic]
0	1	1	1	7			'	7	g	w	G	W			[graphic]	[graphic]	G	W	[graphic]	[graphic]
1	0	0	0	8	BS		(	8	h	x	H	X			[graphic]	[graphic]	H	X	[graphic]	[graphic]
1	0	0	1	9			)	9	i	y	I	Y			[graphic]	[graphic]	I	Y	[graphic]	[graphic]
1	0	1	0	A	NL	SUB	*	:	j	z	J	Z			[graphic]	[graphic]	J	Z	[graphic]	[graphic]
1	0	1	1	B		ESC	+	;	k	{	K	[graphic]			[graphic]	[graphic]	K	[graphic]	[graphic]	[graphic]
1	1	0	0	C			,	<	l	¥	L	[graphic]			[graphic]	[graphic]	L	[graphic]	[graphic]	[graphic]
1	1	0	1	D	CR		−	=	m	}	M	[graphic]			[graphic]	[graphic]	M	[graphic]	[graphic]	[graphic]
1	1	1	0	E	SO		.	>	n	↑	N	[graphic]			[graphic]	[graphic]	N	[graphic]	[graphic]	[graphic]
1	1	1	1	F	SI		/	?	o	←	O	[graphic]			[graphic]	[graphic]	O	[graphic]	[graphic]	[graphic]

UP MODE Code Table

D7					0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
D6					0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
D5					0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
D4					0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
D3	D2	D1	D0		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0		POS	SP	ø	@	P	[graphic]	[graphic]			SP	[graphic]	[graphic]	[graphic]	SP	[graphic]
0	0	0	1	1		CURSOR DOWN	'	1	A	Q	[graphic]	[graphic]		CURSOR UP	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]
0	0	1	0	2		SVS ON	"	2	B	R	[graphic]	[graphic]		REV OFF	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]
0	0	1	1	3			#	3	C	S	[graphic]	[graphic]			[graphic]	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]
0	1	0	0	4			$	4	D	T	[graphic]	[graphic]			[graphic]	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]
0	1	0	1	5			%	5	E	U	[graphic]	[graphic]			[graphic]	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]
0	1	1	0	6			&	6	F	V	[graphic]	[graphic]			[graphic]	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]
0	1	1	1	7			'	7	G	W	[graphic]	[graphic]			[graphic]	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]
1	0	0	0	8	BS		(	8	H	X	[graphic]	[graphic]			[graphic]	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]
1	0	0	1	9			)	9	I	Y	[graphic]	[graphic]			[graphic]	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]
1	0	1	0	A	NL	SUB	*	:	J	Z	[graphic]	[graphic]			[graphic]	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]
1	0	1	1	B		ESC	+	;	K	{	[graphic]	[graphic]			[graphic]	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]
1	1	0	0	C			,	<	L	¥	[graphic]	[graphic]			[graphic]	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]
1	1	0	1	D	CR		−	=	N	}	[graphic]	[graphic]			[graphic]	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]
1	1	1	0	E	SO		.	>	O	↑	[graphic]	[graphic]			[graphic]	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]
1	1	1	1	F	SI		/	?	P	←	[graphic]	[graphic]			[graphic]	[graphic]	[graphic]	[graphic]	[graphic]	[graphic]

Mappa memoria relativa al colore

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
38400																						
38422																						
38444																						
38466																						
38488																						
38510																						
38532																						
38554																						
38576																						
38598																						
38620																						
38642																						
38664																						
38686																						
38708																						
38730																						
38752																						
38774																						
38796																						
38818																						
38840																						
38862																						
38884																						

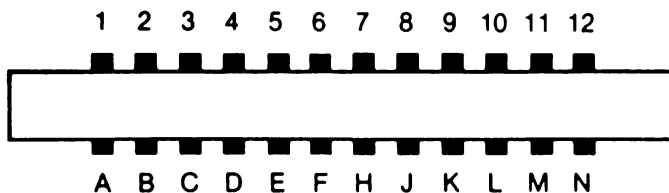
Porta giochi

PIN #	TYPE
1	JOYØ
2	JOY1
3	JOY2
4	JOY3
5	POT Y
6	LIGHT PEN
7	+5V
8	GND
9	POT X

Combinazioni di colori sfondo-bordo

	BORDER							
SCREEN	BLK	WHT	RED	CYAN	PUR	GRN	BLU	YEL
BLACK	8	9	10	11	12	13	14	15
WHITE	24	25	26	27	28	29	30	31
RED	40	41	42	43	44	45	46	47
CYAN	56	57	58	59	60	61	62	63
PURPLE	72	73	74	75	76	77	78	79
GREEN	88	89	90	91	92	93	94	95
BLUE	104	105	106	107	108	109	110	111
YELLOW	120	121	122	123	124	125	126	127
ORANGE	136	137	138	139	140	141	142	143
LT. ORANGE	152	153	154	155	156	157	158	159
PINK	168	169	170	171	172	173	174	175
LT. CYAN	184	185	186	187	188	189	190	191
LT. PURPLE	200	201	202	203	204	205	206	207
LT. GREEN	216	217	218	219	220	221	222	223
LT. BLUE	232	233	234	235	236	237	238	239
LT. YELLOW	248	249	250	251	252	253	254	255

NOTA	VALORE	NOTA	VALORE
DO	135	SOL	215
DO#	143	SOL#	217
RE	147	LA	219
RE#	151	LA#	221
MI	159	SI	223
FA	163	DO	225
FA#	167	DO#	227
SOL	175	RE	228
SOL#	179	RE#	229
LA	183	MI	231
LA#	187	FA	232
SI	191	FA#	233
DO	195	SOL	235
DO#	199	SOL#	236
RE	201	LA	237
RE#	203	LA#	238
MI	207	SI	239
FA	209	DO	240
FA#	212	DO#	241

Valori approssimativi da inserire per ottenere le varie note.

Comandi relativi ai Generatori Sonori	Valori possibili di X	Funzioni
POKE 36878, X	da 0 a 15	Setta volume
POKE 36874, X	da 128 a 255	1° generatore
POKE 36875, X	da 128 a 255	2° generatore
POKE 36876, X	da 128 a 255	3° generatore
POKE 36877, X	da 128 a 255	Generatore

PIN #	TYPE	NOTE	PIN #	TYPE
1	GND		A	GND
2	+5V	100mA MAX.	B	CB1
3	$\overline{\text{RESET}}$		C	PB0
4	JOY0		D	PB1
5	JOY1		E	PB2
6	JOY2		F	PB3
7	LIGHT PEN		H	PB4
8	CASSETTE SWITCH		J	PB5
9	SERIAL ATN IN		K	PB6
10	+9V	100mA MAX.	L	PB7
11	GND		M	CB2
12	GND		N	GND

Giocare con il VIC 20 è divertente; capire come si possono realizzare gli effetti sonori, di grafica e di colore che rendono piacevoli i giochi è istruttivo.
Questo libro è un invito a imparare divertendosi; prima giocate, ma poi date un'occhiata ai listati dei programmi. Imparerete alcune tecniche semplici ma raffinate per raggiungere gli effetti più spettacolari, e questo vi servirà senz'altro quando dai giochi vorrete passare ad applicazioni ... "serie".
Vi accorgerete che il pregio di questa pubblicazione è la varietà dei giochi presentati, scelti con lo scopo di coprire nel modo più completo possibile tutta la gamma di prestazioni che questo (ormai celebre) calcolatore, vi offre.

L. 10.000

Cod. 563 D ISBN 88-7056-028-7

94

SINFONIA PER UN COMPUTER VIC 20

Tim Hartnell

GRUPPO EDITORIALE JACKSON